Key Bridge Global LLC

Date:	2011-10-18
Document reference number:	KB-WSIF-02r00
Version:	2.0r00
Status:	DRAFT
Category:	Implementation Standard
Editor:	Jesse Caulfield

Implementation Standard for White Space Operations

Wireless Service Information Format (WSIF)

Document type:	**Commercial Standard**
Document subtype:	**Commercial**
Document stage:	**DRAFT**
Document language:	**English**

Key Bridge Global LLC

1600 Tysons Blvd., Suite 1100
McLean, VA 22102, USA

Phone: +1 (703) 414-3500
http://KeyBridgeGlobal.com

1 Important Notices

1.1 Copyright

1.2 Amendments and Review

This document describes a commercial implementation strategy for white space administration and is not an official Standard. While Key Bridge is the author, this document includes and has been developed through a collaborative process and incorporates suggestions and editorial advice from many contributors.

The ultimate intent of the authors is to evolve this document, mature its implementation and provide it as a formal contribution such that it may become an Implementation Specification. Nevertheless, while the material in this document has been carefully reviewed, it remains subject to change without notice and should not be referred to as a formal Standard or Specification.

The authors welcome your feedback. Recipients of this document are invited to submit their comments, and if applicable, notification and supporting documentation, to Key Bridge Global by email to documents@keybridgeglobal.com or to the following mailing address:

Key Bridge Global LLC
1600 Tysons Blvd., Suite 1100
McLean, VA 22102, USA

1.3 No Warranty

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED OR OTHERWISE, INCLUDING WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, PERFORMANCE, QUALITY OR NON-INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. KEY BRIDGE FURTHER DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS.

KEY BRIDGE SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUES OR LOST PROFITS THAT MAY RESULT FROM THE USE OF THESE MATERIALS, WHICH IS AT YOUR SOLE RISK AND EXPENSE. THROUGH YOUR USE YOU AGREE TO INDEMNIFY AND HOLD KEY BRIDGE HARMLESS FROM ANY AND ALL DAMAGES RESULTING FROM YOUR USE.

2 Table of Contents

3 Introduction

This document specifies computer software object models and their corresponding XML encoding that collectively describe the core **Wireless Service Information Format (WSIF)**. The WSIF is intended as an information formatting scheme to support other related computer software frameworks and implementations for the presentation, validation, persistence and communication of wireless service descriptions.

The purposes of this document are to:

- Provide a generalized machine and human-readable description of wireless communication services, infrastructure and supporting details
- Provide a standardized data model and information encoding scheme to more efficiently support cross-domain spectrum administration and frequency coordination
- Provide a standardized method to describe accurate, precise and consistently formatted Geo-location information of transmitting and receiving stations and devices
- Provide a complete and extensible description of wireless transmitter and receiver characteristics to assist network planning and wireless service coexistence
- Support the implementation of white space administration in the TV broadcast bands through standardized description and secure distribution of geographic contour information and supporting license details
- Enable the convenient storage, retrieval and long-term archival of wireless service records, frequency reservations and other related information

The WSIF specification further provides a framework within which the operational characteristics of any conventional wireless service consisting of one or more transmitters and one or more receivers may be described. While a great variety of wireless service types, from simple point-to-point microwave to complex multi-node mesh systems exist, these can all be described using the WSIF and varying the available parameters to accommodate unique configurations and validation requirements. The WSIF is designed to accommodate the description of most any generalized wireless service: including licensed and unlicensed stations and incorporating multiple transmitters and receivers in any combination found in the real-world. A WSIF record may also include geographic location, polygons describing areas of operation, license details, equipment and antenna parameters, and identification of the regulatory operating environment. The WSIF software object model and XML schema conveniently describe all TV-Band white space services identified for operation under FCC Rules.

4 Approach

In developing the Wireless Service Information Format (WSIF) specification the authors have attempted to incorporate and build upon other open and mature standards wherever possible. In cases where no satisfactory standard, data model or representation could be found the authors have followed best and standard practices whenever a new, original data model is required.

In some instances an existing standard representation required modification to establish FCC Rules-compliance, and in these cases the original specification was kept wholly intact with the required modification added in a manner matching the original standard's organizing strategy.

WSIF is intended to support white space operations within the TV Bands and beyond, and incorporates features and conventions necessary to accommodate the following necessary functions:

- Administrator Synchronization

White Space Rules describe the minimum set of information that must be contained within a white space database and also require that multiple administrators exchange this information at least once every 24 hours.[1] WSIF supports all inter-administrator information exchange as required by the Rules and also needed for practical implementation of White space operations.

- Content Verification

The wireless service information format described in this document may be used to provide standardized database queries and results format for the confirmation and satisfaction of requirement § 15.715(a).

- Distributed Functionality

The wireless service information format may carry all information contained within FCC database records through its enumerated attributes and elements and support for extensions.

- Non-TVBD Services

WSIF may also be used as a basis to provide general contextual information about all types of wireless transmitters and services, and may serve as a foundation for extending white space operations into new frequency bands.

With regard to definitions and structure, this specification makes the following fundamental assumptions:

- All information is encoded and transmitted using Extensible Markup Language (XML).

[1] See 47 CFR § 15.713(b)(2)

5 How to read this document

XML Schema files (.xsd) are regular text files which can be viewed using any text editor, a web browser, or dedicated XML tools.

WSIF XML schema includes descriptions of objects, compound objects, object parameters and properties. XML refers to these various components with its own syntax of simple types, complex types, attributes and elements. The corresponding software and XML labels are listed below.

Software Label	Description	XML Label
Object	A simple object	Simple Type
Compound Object	An object that contains other objects	Complex Type
Parameter	An object parameter	Attribute
Property	An object property	Element

Complex elements representing software objects are described using graphical illustrations of the XML Schema. See *Appendix: SFA Geometry Class Hierarchy and* Encoding for a complete explanation of how to read the provided XML and diagrams.

Briefly, diagrams are created according to the following convention:

- Attributes of the current Type are grouped inside a box titled “attributes”
- Optional element or attribute are drawn using a dashed line
- Required elements and attributes are drawn in a solid box
- Element lists and repeatable attributes are shown in shadowed boxes with their permitted numbers of occurrence indicated underneath

5.1.1 Optional vs. Required

Object values (XSD attributes and elements) may be declared optional or required. Optional values may be either declared null or omitted from the exchanged XML.

WSIF software object (complex type) illustrations in this document are presented with corresponding XML definitions, which are authoritative. Each complex type is also presented with an enumerated list of its elements and attributes with their type and documentation describing their content and function.

Within each element and attribute table, the respective element or attribute name is formatted to indicate its required status: labels with normal text are required (mandatory), while *italic* text labels are optional.

5.2 Validity

Data elements that contain a dataset identifier (e.g. a date, an integer or a decimal number) must be properly formatted. XML messages that are not well-formed (e.g. not compliant with XML standards) or not valid (e.g. not compliant with the WSIF Schema definition and XML eXtensible Stylesheet Language Transformation (XSLT)) will be rejected.

5.3 Reading the Diagrams

Some diagrams that appear in this specification are presented using the Unified Modeling Language (UML) static structure diagram.

Most diagrams that appear in this document are presented using an XML schema notation defined by the XMLSpy commercial software product. The provided XML schema diagrams are for illustrative purpose only although every attempt has been made to ensure the diagrams accurately reflect the accompanied schema definitions.

An XMLSpy diagram graphically represents the contents of XML files and XSD schemas.[2] In the context of this document the XSD text shall serve to authoritatively describe the XML representation of all software objects.

In XML software object components are referred as elements, which if complex may contain simple (e.g. String, Double, Float, Boolean) and other complex (e.g. List, Map, Object) parameters. Elements (i.e. objects) are shown graphically via a *content model*, which represents the structure and contents of an XML element (or when un-marshaled, as a software object), and is rendered in XMLSpy as a horizontal tree, as shown in *Figure 1*.

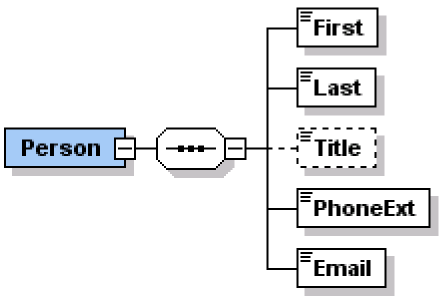
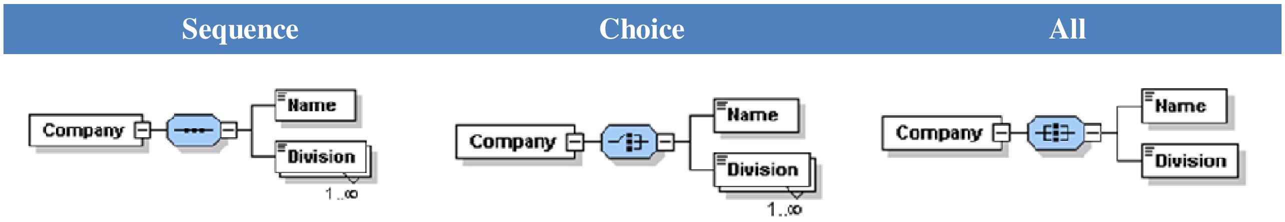


Figure 1: A typical XML element rendering

Within a composite content model a *compositor* defines the order in which child elements occur. There are three compositors available: *sequence*, *choice*, and *all*. These are illustrated in *Figure 2*.


Sequence

Choice

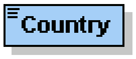
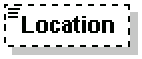
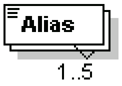
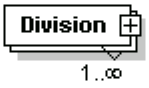
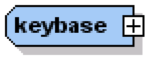
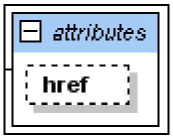
All

Figure 2: Rendering strategies for available XML compositors

[2] For a complete description see Altova XMLSpy® 2011 Enterprise Edition online documentation at http://manual.altova.com/XMLSpy/spyenterprise/

5.3.1 Diagram Components

The graphical representation of a content model's components provides detailed information about the component's type and properties. Components employed in this document include the following types:

Component	Graphic	Description
Mandatory single element		A solid-border rectangle indicates an element is required. No number range indicates a single element (i.e. minOcc=1 and maxOcc=1). The name of the element shown is *Country*.
Optional single element		The dashed-border rectangle indicates an optional element. The absence of a number range indicates a single element (i.e. minOcc=0 and maxOcc=1). The name of the element shown is *Location*.
Mandatory multiple element		The solid-border rectangle indicates the element is required, and the number range 1..5 signifies that minOcc=1 and maxOcc=5. This element name is *Alias*.
Mandatory multiple element containing child elements		The solid-border rectangle indicates the element is required, while the number range 1..infinity means that minOcc=1 and maxOcc=unbounded. The plus sign indicates this element is a complex type with embedded content (i.e. at least one element or attribute child). This element name is *Division*.
Complex type		The irregular hexagon with a plus sign indicates a global complex type. Within the context of this document, global complex types are always employed as the data type of an element.
Attributes		Complex type objects may include attributes, which are indicated with the word 'attributes' shown italics and enclosed within a solid rectangle. Each attribute is shown in its own rectangle which may have a dashed border if the attribute is optional or a solid border if the attribute is required.

5.4 Naming

Lower camel case is used for all WSIF naming. Mixed upper- and lower-case are permitted in any data entry and should be consistently maintained throughout any implementations, except for any pre-defined enumerated codes, which must be exchanged exactly as represented.

5.5 Units

The formats and units in the WSIF attribute definitions are the units of exchanged data. With regard to definitions and structure the WSIF specification makes the following fundamental assumptions for units of power, frequency and geodetic datum.

5.5.1 Power

WSIF power is exchanged in units of dBW in accordance with ITU treaty procedures, with power in dBW is calculated as:

$$P_{dBW} = 10 \log_{10} P_W$$

where

P_{dBW} = Power in dBW

P_{dBW} = Power in Watt

5.5.2 Frequency

WSIF frequency values are always noted in MHz. WSIF follows common practice and ITU convention in which a frequency value is formatted with between zero to five decimal places. Accordingly, all WSIF frequencies are measured and communicated in units of MHz, where the minimum allowable unit of frequency is 1.0 Hz. Frequency units are not otherwise explicitly declared.

5.5.3 Geodetic datum

All WSIF geo-location references are measured and communicated using the World Geodetic System of 1984 (WGS84) geodetic datum unless otherwise noted. [3]

[3] See United States Department of Defense; *DoD WGS-1984 – Its Definition and Relationships with Local Geodetic Systems*; Washington, D.C.; 1985; Report AD-A188 815 DMA; 6127; 7-R-138-R; CV, KV;

5.6 Element De-Duplication

WSIF is intended to support the complete definition, storage and exchange of wireless service information. WSIF data exchanges should be fully self-contained and not make external reference to information that may not be available to the recipient.

Objects that appear multiple times may be identified by XPATH reference to avoid duplication of transmitted information. However an object reference is only allowed if the referred entity has already been transmitted or defined in the message. [4]

For example, within a WSIF *wirelessService* several *station* elements may refer to the same *antenna*. In this case the first *station* must fully describe the *antenna* and subsequent stations may refer to the already described *antenna* element via XPATH location.

5.7 Use of XPATH

In general, the following limitations SHOULD be applied when incorporating XPATH expressions within the WSIF:

- Functions MAY NOT be used. Instead, use fixed index predicates e.g. [1] or [@name="MODE 1"]
- XPATH expressions MUST start with "/" to indicate that the path is relative to the dataset root
- XPATH must select a single node (element or attribute)

5.8 Examples

See *Appendix: Example WSIF Implementations.*

5.9 Prohibited and restricted characters

XML is a tag-based text format with reserved characters for tag and notation delimiters. The following characters are not allowed. Instead, the URL-encoded text should be used as follows:

Prohibited character	URL-encoded string
<	&lt;
>	&gt;
"	&quot;
&	&amp;

It is the software sending party's responsibility to ensure that characters are correctly encoded. Incorrectly encoded XML messages should be discarded with error notification to the sender.

[4] See W3C *XML Path Language (XPath) Version 1.0,* Copyright © 1999 W3C® (MIT, INRIA, Keio) at http://www.w3.org/TR/xpath/

6 Terms and Definitions

Term	Definition
Complex Type	XML label syntax for a computer software object that incorporates other software objects.
Coordinate System (CS)	According to ISO19111, a set of (mathematical) rules for specifying how coordinates are assigned to points. In this document, a Coordinate System is extended to be defined as a set of axes with which location and orientation can be defined.
Datum	Undefined in ISO 19111. Defined here as a means of relating a coordinate system to the real world by specifying the physical location of the coordinate system and the orientation of the axes relative to the physical object. For a geodetic datum, the definition also includes a reference ellipsoid that approximates the surface of the planetary body.
Location	A point or extent in space relative to a coordinate system. For point-based systems, this is typical expressed as a set of n-dimensional coordinates within the coordinate system.
Marshalling, Unmarshalling	Marshalling is the process of transforming a software object to a serialized data format for storage or transmission. Marshalling is typically used when data must be distributed or exchanged between different computer programs or systems. In the context of this document, marshalling is the process of converting an in-memory software component (i.e. a Java object) into an XML message. The reverse process of marshalling is unmarshalling, which in the context of this document is the conversion of an XML message into a software component. Marshalling and unmarshalling may be synonymously referred to as serialization and de-serialization, respectively.
Serialize, De-serialize	See Marshalling, Unmarshalling.
Universally Unique Identifier	A universally unique identifier (UUID) is an identifier standard used in software construction, standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE). The intent of UUIDs is to enable distributed systems to uniquely identify information without significant central coordination. In this context the word unique should be taken to mean "practically unique" rather than "guaranteed unique". UUIDs are documented as part of ISO/IEC 11578:1996 *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC)* and more recently in ITU-T Rec. X.667 | ISO/IEC 9834-8:2005. The IETF has published Standards Track *RFC 4122* that is technically equivalent with ITU-T Rec. X.667 | ISO/IEC 9834-8. A UUID is a 16-byte (128-bit) number. The number of theoretically possible UUIDs is therefore about 3 × 10^{38}. In its canonical form, a UUID consists of 32 hexadecimal digits, displayed in 5 groups separated by hyphens, in the

	form 8-4-4-4-12 for a total of 36 characters (32 digits and 4 hyphens). For example: 4aa9aac8-acf5-11e0-8c31-0024211c6fd1
URL-Encoding	URL encoding converts characters into a format that can be transmitted over the Internet. URLs can only be sent over the Internet using the ASCII character-set. Since URLs often contain characters outside the ASCII set, the URL has to be converted into a valid ASCII format. URL encoding replaces unsafe ASCII characters with a "%" followed by two hexadecimal digits. URLs cannot contain spaces. URL encoding normally replaces a space with a + sign.

6.1 Symbols (and abbreviated terms)

Acronym	Definition
CFR	Code of Federal Regulations
CRS	Coordinate Reference System
GPS	Global Positioning System
ISO	International Organization for Standardization
OSI	Open System Interconnection
uom	Unit(s) of measure
UML	Unified Modeling Language
URI	Universal Resource Identifier
URN	Universal Resource Name
WGS84	World Geodetic System 1984
XML	eXtensible Markup Language
xs:*	Namespace prefix for XMLSchema
CFR	Code of Federal Regulations
UUID	Universally Unique Identifier

7 Protocol Versions

The following protocol versions should be applied when using the specifications described in this document. The revision number may be optionally noted but is not required.

Object	Protocol Version	Revision Number
protocolVersion	2.0	0
securityVersion	2.0	0

7.1 Protocol Security

This specification describes data formatting and encoding strategies to enable standardized messaging of WSIF information.

The Format for White Space Messaging (WSM) is composed of standard XML and is therefore compatible with many commonly available message encryption and transport security technologies.

However, actual message encryption, counter-party authentication, communications transport and other security aspects of a working system are beyond this document's scope. Information assurance and protocol security is the responsibility of each implementing party.

8 The WSIF Object Model Overview

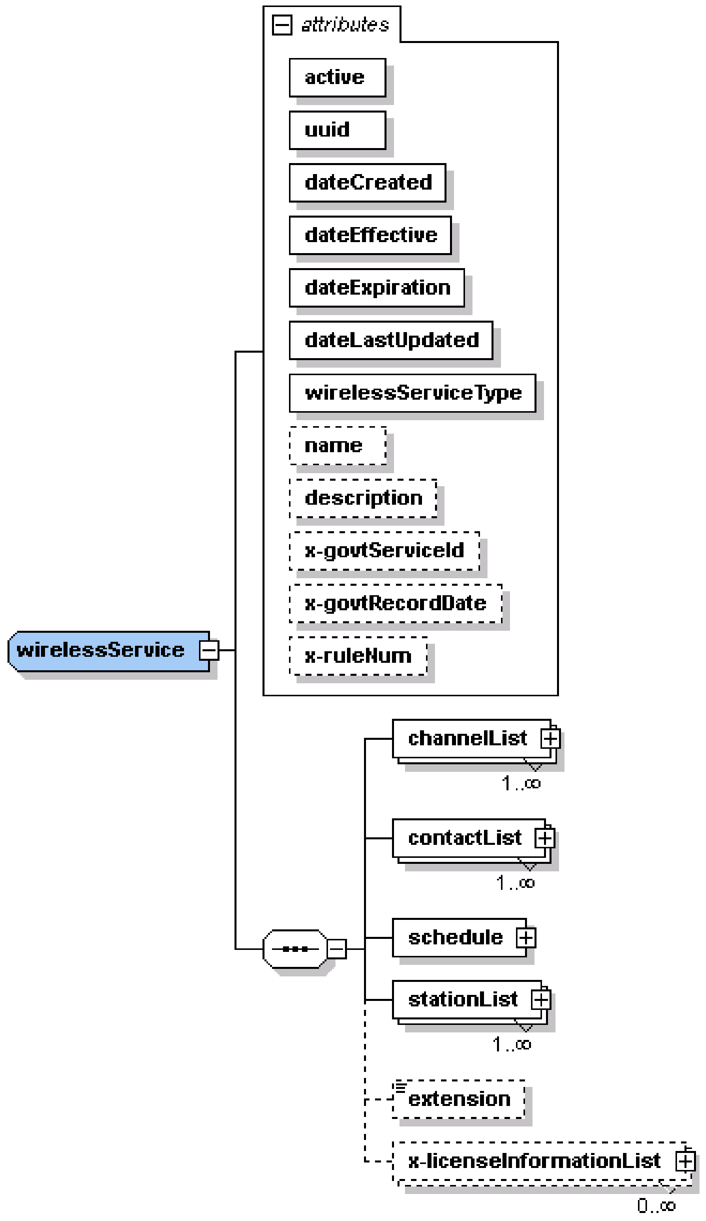


Figure 3: A partially expanded *wirelessService* complex type showing attributes and elements. The *wirelessService.channelList*, *contactList, schedule* and *stationList* elements are required (solid box) while the *extension* and *x-licenseInformationList* elements (dashed box)) are optional. The service's *active*, *uuid*, *effectiveDate*, *expirationDate*, *wirelessServiceType, createDate* and *lastUpdated* attributes are required while the other attributes are optional. Also, the number of allowed list elements is indicated as a subscript to the various list boxes.

A WSIF *wirelessService*, showing its top-level components, is illustrated in *Figure 3*.

The WSIF supports the description of various types of wireless services through configuration of the indicated attributes and elements.

A key attribute required for WSIF implementation is the *wirelessServiceType*, which authoritatively describes the service and which may be used to determine which and how many elements and attributes are required.

Other important WSIF Types are

- *channel*: which authoritatively describes a range of frequencies
- *contact:* which provides a standard container for individual and organization contacts
- *station*: which provides a standard container for a transmitting facility or device and their respective transmitting parameters, geographic location and geographic protected contour or service area
- *license*: which provides a standard description of Government issued licenses

Also important are the primitive value definitions, for which the WSIF specification uses simple data types described in the W3C *XML Schema Part 2: Data types Second Edition.* [5]

[5] Available online at http://www.w3.org/TR/xmlschema-2/#built-in-datatypes

8.1 address

A WSIF *address* is a standardized container for physical (e.g. mailing) address information.

WSIF *address* attributes and names are chosen to closely match existing US Government databases (e.g. Postal Service, FCC, NOAA, NASA) and to also support international address content requirements.

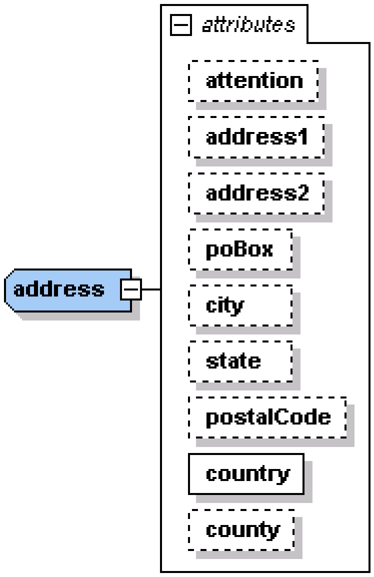


Figure 4: An address complex type

8.1.1 Definition

```
<xs:complexType name="address">
        <xs:attribute name="attention"              type="xs:string"/>
        <xs:attribute name="address1"               type="xs:string"/>
        <xs:attribute name="address2"               type="xs:string"/>
        <xs:attribute name="poBox"                  type="xs:string"/>
        <xs:attribute name="city"                   type="xs:string"/>
        <xs:attribute name="county"                 type="xs:string"/>
        <xs:attribute name="state"                  type="xs:string"/>
        <xs:attribute name="postalCode"             type=“xs:string"/>
        <xs:attribute name="country"                type=“xs:string" use="required"/>
    </xs:complexType>
```

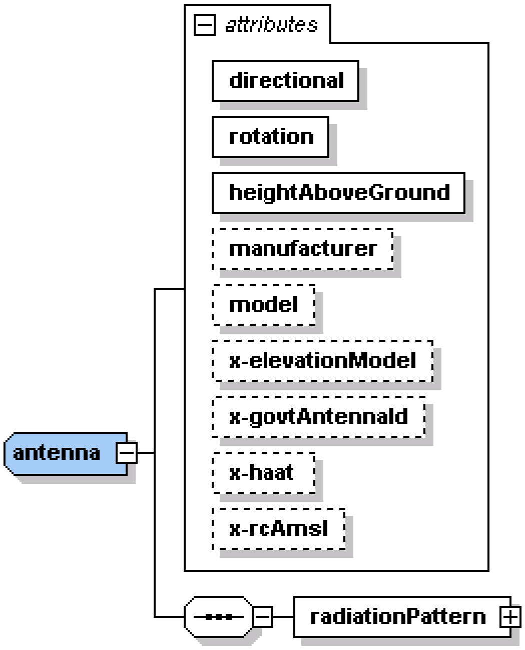
8.1.2 Attributes

Attribute	Type	Documentation
attention	xs:string	The person or role to whom correspondence should be addressed
address1	xs:string	The street address
address2	xs:string	The suite number or mail stop
poBox	xs:string	A post box if specified
city	xs:string	The city
state	xs:string	The state or administrative area
postalCode	xs:string	The alphanumeric postal code (e.g. zip code for US destinations)
country	xs:string	An ISO 3166-1 alpha-2 country code
county	xs:string	The city's political incorporation

8.1.3 Validation

The *country* attribute is required.

8.2 antenna



A WSIF *antenna* holds required antenna parameters like pointing and elevation information plus the radiation pattern.

The WSIF *antenna.radiationPattern* element describes the directional gain of the antenna with encoded azimuthal field values.

Optional *antenna* parameters are useful for the complete description of a transmitting or receiving antenna.

Figure 5: An antenna complex type

8.2.1 Definition

```
<xs:complexType name="antenna">
        <xs:sequence>
            <xs:element name="radiationPattern"    type="radiationPattern"/>
        </xs:sequence>
        <xs:attribute name="directional"           type="xs:boolean" use="required"/>
        <xs:attribute name="rotation"              type="xs:float" use="required"/>
        <xs:attribute name="heightAboveGround"     type="xs:float" use="required"/>
        <xs:attribute name="manufacturer"          type="xs:string"/>
        <xs:attribute name="model"                 type="xs:string"/>
        <xs:attribute name="x-elevationModel"      type="xs:string"/>
        <xs:attribute name="x-govtAntennaId"       type="xs:int"/>
        <xs:attribute name="x-haat"                type="xs:float"/>
        <xs:attribute name="x-rcAmsl"              type="xs:float"/>
    </xs:complexType>
```

8.2.2 Elements

Element	Type	Documentation
radiationPattern	radiationPattern	A radiation pattern representing the directional (azimuth) dependence of the strength of the radio signal from the antenna.

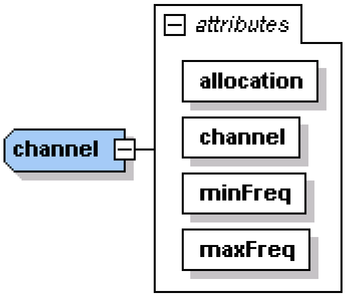
8.2.3 Attributes

Attribute	Type	Documentation
directional	xs:boolean	Indicates whether the antenna is directional (true) or non-directional (false).
rotation	xs:float	Indicates the offset in degrees azimuth [0, 360] from true North that the antenna radiation pattern should be rotated.
heightAboveGround	xs:float	The antenna radiation center height above ground level. *Key Bridge implementation* Personal/Portable (Mode II) TV Band devices may leave this field empty. For Mode II TV Band devices, if null, a default value of 2m is assumed.
manufacturer	xs:string	The antenna manufacturer.
model	xs:string	The antenna product model.
x-elevationModel	xs:string	*Key Bridge implementation* The digital elevation model used to calculate this *antenna* object's HAAT (*x-haat*) and rcAMSL (*x-rcAmsl*) values.
x-govtAntennaId	xs:int	*Key Bridge implementation* A reference to the antenna ID record within the FCC CDBS.
x-haat	xs:float	*Key Bridge implementation* The antenna height above average terrain as reported in the FCC CDBS.
x-rcAmsl	xs:float	*Key Bridge implementation* The antenna radiation center above mean sea level as reported in the FCC CDBS.

8.2.4 Validity

If a non-directional antenna is indicated by setting the *directional* attribute to false then the *rotation* attribute should be set to zero (0) and the *radiationPattern* element may be optionally sent as an empty element by the sending party. The *radiationPattern* is not nullable.

8.3 channel



A WSIF *channel* describes a fully qualified and canonical frequency range.

WSIF *channel* object definitions support positive definitions of colloquial channel identifiers (e.g. TV channel 24) through identification of the authorizing regulatory definition and the TV channel's frequency range.

Figure 6: A channel complex type

8.3.1 Definition

```
<xs:complexType name="channel">
        <xs:attribute name="allocation"          type="xs:string" use="required"/>
        <xs:attribute name="channel"             type="xs:float" use="required"/>
        <xs:attribute name="minFreq"             type="xs:double" use="required"/>
        <xs:attribute name="maxFreq"             type="xs:double" use="required"/>
</xs:complexType>
```

8.3.2 Attributes

Attribute	Type	Documentation
allocation	xs:string	A dot-delimited reverse domain encoded description of the frequency allocation defined according the following strategy: [country].[regulator].[allocation].[band range] For example, the UHF-high block allocation of TV channels 38 to 51 within the United States is identified as "us.fcc.broadcast.614-698". [6]
channel	xs:float	The colloquial channel number Note: A FLOAT number type is used to accommodate future sub-channelization. For the avoidance of doubt channel numbers ending in zero shall be interpreted to represent a whole channel. i.e. float value channel 38.0 is equivalent to integer-value channel 38.
minFreq	xs:double	The minimum (or start) frequency of the indicated channel in MHz
maxFreq	xs:double	The maximum (or end) frequency of the indicated channel in MHz

[6] See *Table of Frequency Allocations* as published by the Federal Register at 47 C.F.R. 2.106 for the example given. Also see *Appendix: Enumerated Codes* for a list of allocations, TV channels and frequencies assigned for US white space implementation.

8.3.3 Validity

WSIF *channels* provided by white space administrators SHOULD include a proper and complete *allocation* reference. WSIF *channel* objects provided by an administrator without an *allocation* attribute SHOULD NOT be used for the purposes of initiating a wireless transmission.

WSIF *channels* provided by white space devices may provide an empty (but not null) *allocation* attribute only if their regulatory environment can be authoritatively determined by other means (e.g. by providing certification information that authorizes the device's current operating location, or by other methods.)

8.4 contact

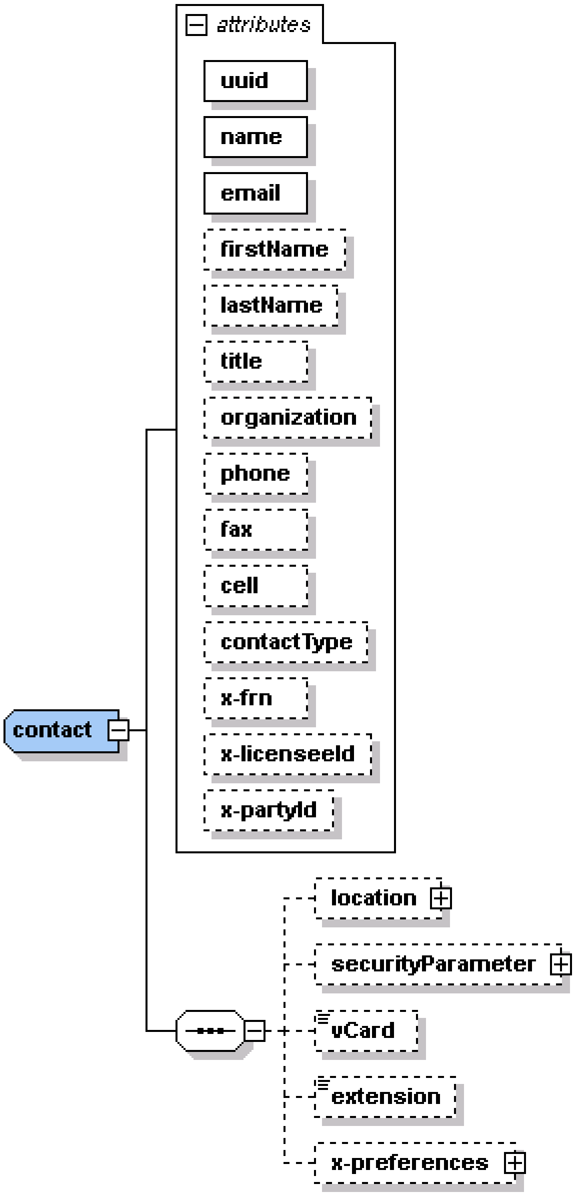


Figure 7: A contact complex type

The WSIF *contact* represents a generalized container for individual (person) and company (organization) contact information. The WSIF *contact* is derived from the IETF vCARD specification with minor extensions to accommodate useful information provided by Government databases or information otherwise required for white space implementation by Rule.

A WSIF *contact* object organizes information necessary to describe an organization or person, the organization or person's role, their security credentials, and how they may be reached via a location object which captures mailing addresses.

8.4.1 Definition

```
<xs:complexType name="contact">
        <xs:sequence>
            <xs:element name="location"          type="location" minOccurs="0"/>
            <xs:element name="securityParameter" type="securityParameter" minOccurs="0"/>
            <xs:element name="vCard"             type="xs:string" minOccurs="0"/>
            <xs:element name="extension"         type="xs:string" minOccurs="0"/>
            <xs:element name="x-preferences"     type="preferences" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="uuid"              type="xs:string" use="required"/>
        <xs:attribute name="name"              type="xs:string" use="required"/>
        <xs:attribute name="email"             type="xs:string" use="required"/>
        <xs:attribute name="firstName"         type="xs:string"/>
        <xs:attribute name="lastName"          type="xs:string"/>
        <xs:attribute name="title"             type="xs:string"/>
        <xs:attribute name="organization"      type="xs:string"/>
        <xs:attribute name="phone"             type="xs:string"/>
        <xs:attribute name="fax"               type="xs:string"/>
        <xs:attribute name="cell"              type="xs:string"/>
        <xs:attribute name="contactType"       type="xs:string"/>
        <xs:attribute name="x-frn"             type="xs:string"/>
        <xs:attribute name="x-licenseeId"      type="xs:string"/>
        <xs:attribute name="x-partyId"         type="xs:int"/>
    </xs:complexType>
```

8.4.2 Elements

Element	Type	Documentation
location	location	A WSIF *location* object provides the ability to associate a mailing address and geo-location information with a contact.
securityParameter	securityParameter	Each contact (person or organization) may provide security credentials including their public key to enable positive identification, data encryption, and authentication and secure access to resources.
vCard	xs:string	This WSIF *contact* object includes only a subset of all possible values that may be provided from a vCard file. If this *contact* was created from a vCard file then the original vCard may be stored for convenience and future reference.[7]
extension	xs:string	A URL-ENCODED string containing key/value pairs that provides additional information or extends this object.

[7] See IETF RFC 6350: *vCard Format Specification* at http://tools.ietf.org/html/rfc6350

x-preferences	preferences	*Key Bridge implementation* A person or organization may wish to identify certain default handling or workflow preferences, which may be indicates here.

8.4.3 Attributes

Attribute	Type	Documentation
uuid	xs:string	A universally unique identifier (UUID) associated with and permanently assigned to this contact.
name	xs:string	The contact name. *Key Bridge implementation* This mandatory attribute is set according to the following rule: • The concatenation of the "*firstName lastName*" if a person, or • The *email* address if *firstName* AND *lastName* are not available, or • The *organization* attribute if a company, or • The *uuid* attribute if the *firstName* AND *lastName* AND *organization* AND *email* attributes are not available
email	xs:string	A valid email address according to the syntax of RFC 2822.[8] Typical address syntax is of the form "user@host.domain" or "Personal Name <user@host.domain>".
firstName	xs:string	A person's first name.
lastName	xs:string	A person's family (last) name.
title	xs:string	A person's working title.
organization	xs:string	The organization (company) name.
phone	xs:string	Main or direct telephone number encoded as [country code] [phone number]. If country code is omitted (e.g. 10 digits) then 1 (USA) shall be assumed.
fax	xs:string	Fax number.
cell	xs:string	Person's mobile number. This should able to send and receive SMS messages.
contactType	xs:string	An administrator encoded text field used to identify the contact type and its available roles and privileges.
x-frn	xs:string	*Key Bridge implementation* Convenience field to report this contact's FCC registration number.
x-licenseeId	xs:string	*Key Bridge implementation*

[8] See RFC 2822 online at http://tools.ietf.org/html/rfc2822.htm. Specifically, *3.4. Address Specification* and *3.4.1. Addr-spec specification*: "An addr-spec is a specific Internet identifier that contains a locally interpreted string followed by the at-sign character ("@", ASCII value 64) followed by an Internet domain."

		Convenience field to report this contact's FCC database licensee ID number.
x-partyId	xs:int	*Key Bridge implementation* Convenience field to report this contact's *party_id* value recorded in the FCC's CDBS *party* table.

8.4.4 Validity

If the *contact* is a person then the *firstName* and *lastName* attributes are required.

If the *contact* is an organization or company then the *organization* attribute is required.

The following *contact* attributes and elements are required by white space Rule:

For Fixed TVBD registration:[9]

- Either (*firstName* and *lastName*) or *organization*
- *email*
- *location.address*
- *phone*

For Licensed and Unlicensed low-power auxiliary devices:[10]

- Either (*firstName* and *lastName*) or *organization*
- *email*
- *location.address*
- *phone*

Key Bridge implementation

If the *contact* does not have an email address then the *email* attribute is set to a concatenation of either “null-” or “Null-” plus the *uuid* attribute value. This assures WSIF validity while accommodating the importation of records from external (i.e. Government) databases that may not consistently provide a valid email address.

For example: null-89afe673-92a8-40ad-bee9-131d9ea5569c

[9] See 47 CFR 15.713(f)

[10] See 47 CFR 15.713(h)(8) and (9)

8.5 coordinate

The WSIF *coordinate* object is influenced by and extends the W3C *Geolocation API Specification* and the Google *Geolocation API*, which is an implementation of the W3C Geolocation API Specification.[11, 12]

The WSIF coordinate definition supports machine-friendly access to geographical location (position) information.

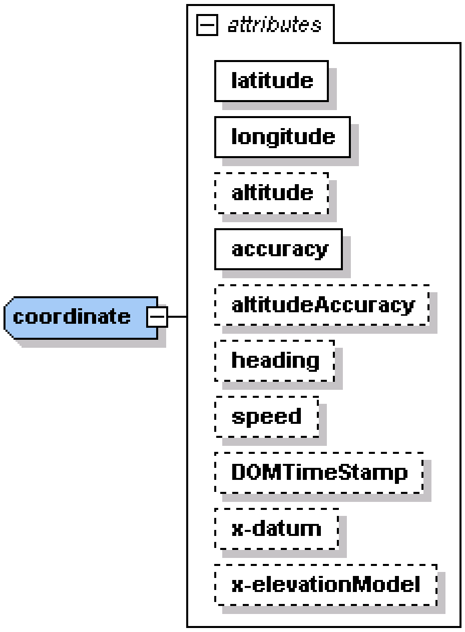


Figure 8: A coordinate complex type

8.5.1 Definition

```
<xs:complexType name="coordinate">
        <xs:attribute name="latitude"          type="xs:double" use="required"/>
        <xs:attribute name="longitude"         type="xs:double" use="required"/>
        <xs:attribute name="altitude"          type="xs:double"/>
        <xs:attribute name="accuracy"          type="xs:double" use="required"/>
        <xs:attribute name="altitudeAccuracy"  type="xs:double"/>
        <xs:attribute name="heading"           type="xs:double"/>
        <xs:attribute name="speed"             type="xs:double"/>
        <xs:attribute name="DOMTimeStamp"      type="xs:long"/>
        <xs:attribute name="x-datum"           type="xs:string"/>
        <xs:attribute name="x-elevationModel"  type="xs:string"/>
    </xs:complexType>
```

[11] See Google Gears (Deprecated) *Geolocation API*, Copyright ©2011 Google at http://code.google.com/apis/gears/api_geolocation.html#coords. The attribute definitions of this section are Copyright © Google and are reproduced here in accordance with the Google Terms of Service available online at http://www.google.com/accounts/TOS.

[12] See *Geolocation API Specification, W3C Candidate Recommendation 07 September 2010*, Copyright © 2010 W3C, Editor: Andrei Popescu, Google, Inc at http://www.w3.org/TR/geolocation-API/

8.5.2 Attributes

Attribute	Type	Documentation
latitude	xs:double	Latitude in decimal degrees using the World Geodetic System 1984 (WGS84) datum. Latitude value precision is limited to 6 significant digits according to the number format 00.000000 ranging between the values of -90.0 and +90.0.
longitude	xs:double	Longitude in decimal degrees (WGS84) datum. Longitude value precision is limited to 6 significant digits according to the number format 000.000000 ranging between the values of 0.0 and +360.0.
altitude	xs:double	Height in meters (WGS84 datum), or null if not available
accuracy	xs:double	The horizontal accuracy of the position in meters, or null if not available
altitudeAccuracy	xs:double	The vertical accuracy of the position in meters, or null if not available
heading	xs:double	Denotes the direction of travel of the reporting device in degrees counting clockwise relative to the true north. The allowable range of values is 0.0 to 360.0 degrees.
speed	xs:double	Denotes the magnitude of the horizontal component of the reporting device's current velocity measured in meters per second.
DOMTimeStamp	xs:long	Represents the time when the position was acquired and is represented as a DOMTimeStamp [See DOMTIMESTAMP] (typically the system's current time in milliseconds.)
x-datum	xs:string	*Key Bridge implementation* The geodetic datum in which the coordinates are measured. For most GPS-derived coordinates this is “WGS84”, and for FCC-derived coordinates this is “NAD-83”.
x-elevationModel	xs:string	*Key Bridge implementation* The elevation model used to establish the *altitude* and *altitudeAccuracy* attributes of this *coordinate*. Examples of possible elevation models include: SRTM1, SRTM3, NED1, GLOBE, etc.

8.5.3 Elements

The *coordinate* object has no elements.

8.5.4 Validity

Latitude values outside the allowable range should be rejected as invalid. For example, a *longitude* value of +105.0 degrees could indicate an incorrectly encoded longitude or an invalid *latitude* value, either of which is impossible to authoritatively discern programmatically.

Longitude values outside the allowable range may be optionally rejected as invalid or normalized according to a receiving party's discretion. For example, a *longitude* value of -45.0 degrees may be correctly normalized to +270 degrees.

If *heading* information is not available the value of this attribute must be null, and if the reporting device is stationary (i.e. the value of the speed attribute is 0), then the value of the heading attribute must be NaN.

If the *speed* information is not available the value of the *speed* attribute must be null. Otherwise, the value of the speed attribute must be a non-negative real number.

If the x-*datum* attribute is set (not-null) then the *latitude* and *longitude* attributes of this *coordinate* are measured in the indicated datum. If the *datum* attribute is not set (null) then the *latitude* and *longitude* attributes are measured in the WGS84 datum according to the W3C specification. The presence of the *datum* attribute modifies how this WSIF *contact* MUST be interpreted, and is slightly different from the standard definitions of *latitude* and *longitude* as defined in the W3C document.

If the *altitude* was not measured by GPS but rather has been calculated according to a digital elevation model then the elevation model must be indicated in the x-*elevationModel* attribute.

8.6 enumCode

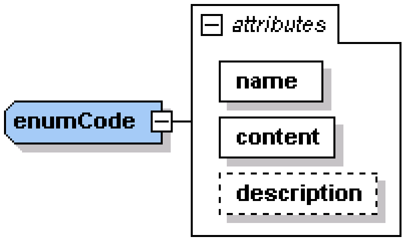


Figure 9: An enumCode complex type

A WSIF *enumCode* is a convenience object for storing and exchanging key-value pairs of defined enumerated codes. WSIF *enumCodes* are intended to enable various WSIF implementations with a convenient means to extend the various WSIF class and type fields as those implementations may require.

When enumerated codes are communicated between parties they must be guaranteed unique, properly named and contain properly formatted content. Including a human-readable description is also recommended.

8.6.1 Definition

```
<xs:complexType name="enumCode">
        <xs:attribute name="name"             type="xs:string" use="required"/>
        <xs:attribute name="content"          type="xs:string" use="required"/>
        <xs:attribute name="description"      type="xs:string"/>
</xs:complexType>
```

8.6.2 Attributes

Attribute	Type	Documentation
name	xs:string	The enumerated code name: a unique string encoded in reverse dot-delimited domain notation that should be human readable and naturally understood. For example, an enumerated code for a white space administrator authorized for operation within the United States could be named: "us.ws.administrator.KEYBRIDGE"
content	xs:string	A URL-encoded set of administrator-defined key-value pairs.[13]
description	xs:string	A human readable description of the code name

[13] See *Terms and Definitions: URL-Encoding*

8.7 geometry

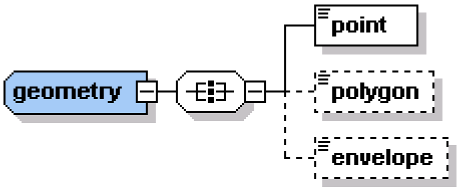


Figure 10: A geometry complex type

A WSIF *geometry* holds geographic and geo-location information. All element values within a WSIF *geometry* are encoded as simple feature access (SFA) well-known text (WKT), a text markup language for representing vector geometry objects on a map, spatial reference systems of spatial objects and transformations between spatial reference systems.[14]

The WSIF specification employs simple feature access (SFA; also called ISO 19125) as its standard for the communication and exchange of all geographic information. SFA is both an OpenGIS and ISO Standard that specifies a common, and more importantly interoperable, storage and communication model of geospatial information like points (e.g. a longitude, latitude pair), lines (e.g. a border), polygons (e.g. a protected service contour) and multi-point (e.g. a wireless microphone registration) configurations using well-known-text (and/or binary) representation.

Adopting SFA as a standard representation of geographic information brings many advantages for the implementation of WSIF including:

- Standards-based association of geographic points, lines and contours with a spatial reference system (a map projection or geodetic datum)
- Mature commercial and free, open-source software implementations and libraries available for most all popular development environments and languages including C, C#, C++, Java, Python, Perl, and others
- Standards-based and interoperable implementations in all popular database systems including Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Informix, sqlite, Derby and others, all of which support the persistence, inquiry and reproduction of SFA-encoded information through their respective geospatial extensions

[14] See Open Geospatial Consortium Inc., *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture Version 1.2.1*, ed. John R. Herring

8.7.1 Definition

```
<xs:complexType name="geometry">
        <xs:all>
            <xs:element name="point"              type="xs:string"/>
            <xs:element name="polygon"            type="xs:string" minOccurs="0"/>
            <xs:element name="envelope"           type="xs:string" minOccurs="0"/>
        </xs:all>
</xs:complexType>
```

8.7.2 Attributes

The *geometry* object as no attributes.

8.7.3 Elements

Each of the elements of the WSIF geometry object are encoded as SFA WKT objects. For more information see *Appendix: SFA Geometry Class Hierarchy and Encoding*.

.

Element	Type	Documentation
point	xs:string	A WKT POINT or optionally WKT MULTIPOINT SFA Geometry implementation.
polygon	xs:string	A WKT POLYGON or optionally WKT MULTIPOLYGON SFA Geometry implementation.
envelope	xs:string	A WKT MULTIPOINT envelope calculated from the *polygon* element of this object.

8.7.4 Validation

The *point* element is required.

The *polygon* element may be required according to various implementations.

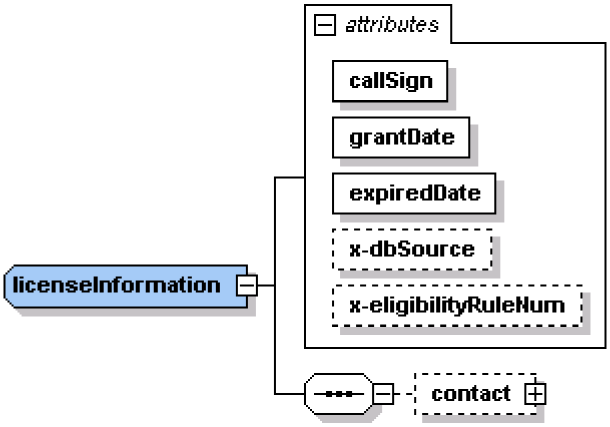
If the *polygon* element is set then the *envelope* element should be set.

Key Bridge implementation

If a *polygon* element is set and the *point* element is not otherwise defined, the *point* element is set to the calculated *polygon* center coordinate.

If a *polygon* element is set the *envelope* element is required but not vice-versa.

8.8 licenseInformation



Government broadcast licenses are typically issued to organizations for commercial wireless service and to individuals for amateur operation.

In the US a license may be uniquely identified by its call sign.

A WSIF *licenseInformation* provides a convenient container to hold and exchange regulatory license information and related details, including the authorized services and licensed entity.

Figure 11: A licenseInformation complex type

8.8.1 Definition

```
<xs:complexType name="licenseInformation">
        <xs:sequence>
            <xs:element name="contact"             type="contact" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="callSign"            type="xs:string" use="required"/>
        <xs:attribute name="grantDate"           type="xs:dateTime" use="required"/>
        <xs:attribute name="expiredDate"         type="xs:dateTime" use="required"/>
        <xs:attribute name="x-dbSource"          type="xs:string"/>
        <xs:attribute name="x-eligibilityRuleNum"  type="xs:string"/>
    </xs:complexType>
```

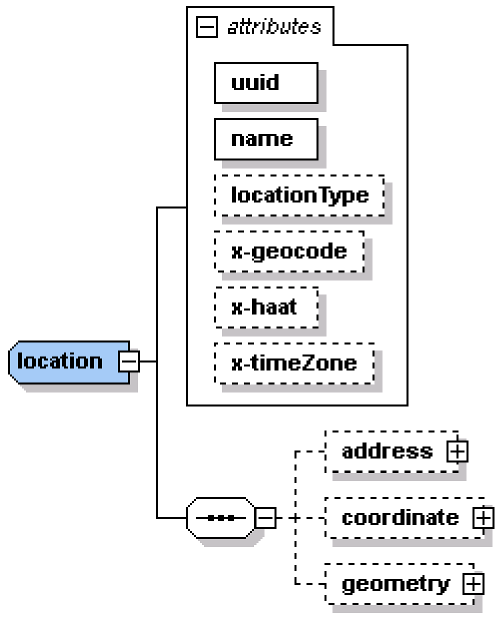
8.8.2 Attributes

Attribute	Type	Documentation
callSign	xs:string	The Government issued call sign (also call name or call letters), a unique alphanumeric designation for a transmitting station.
grantDate	xs:dateTime	The Government license grant date.
expiredDate	xs:dateTime	The Government license expiration date.
x-dbSource	xs:string	*Key Bridge implementation* The database source of this license information identified by its enumerated code name.
x-eligibilityRuleNum	xs:string	*Key Bridge implementation* The regulatory rule number under which the wireless service is licensed to operate. For U.S. white space operation this is a reference to the CFR.

8.8.3 Elements

The *licenseInformation* object has no elements.

8.9 location



A WSIF *location* holds all geographic information used to describe a physical location, an extended area, and motion characteristics.

The WSIF *location* is a compound container that fully describes required aspects of a geographic location, including address, geo-location coordinates and extended geometries.

The WSIF location object is incorporated as an element into several other WSIF objects with different validation requirements. Therefore, except for the mandatory *uuid* and *name* attributes the location validation requirements are context sensitive.

Figure 12: A location complex type

8.9.1 Definition

```
<xs:complexType name="location">
        <xs:all>
            <xs:element name="address"            type="address" minOccurs="0"/>
            <xs:element name="coordinate"         type="coordinate" minOccurs="0"/>
            <xs:element name="geometry"           type="geometry" minOccurs="0"/>
        </xs:all>
        <xs:attribute name="uuid"                 type="xs:string" use="required"/>
        <xs:attribute name="name"                 type="xs:string" use="required"/>
        <xs:attribute name="locationType"         type="xs:string"/>
        <xs:attribute name="x-geocode"            type="xs:string"/>
        <xs:attribute name="x-haat"               type="xs:float"/>
        <xs:attribute name="x-timeZone"           type="xs:string"/>
</xs:complexType>
```

8.9.2 Elements

Element	Type	Documentation
address	address	The mailing address
coordinate	coordinate	The Geo-location coordinates
geometry	geometry	Geo-location and geographic information

8.9.3 Attributes

Attribute	Type	Documentation
uuid	xs:string	A universally unique identifier (UUID) associated with and permanently assigned to this *location*.
name	xs:string	A human readable name or label that may be used to identify this location. A useful hint is to use a memorable place name as might be represented on a map (e.g. “Empire State Building”).
locationType	xs:string	A descriptor string used to classify and organize locations. *Key Bridge implementation* If the location is derived from another database source, this attribute is a dot-delimited string used to identify this *location* type and its source. An example value for this attribute is “us.fcc.cdbs.stationClass.CDT”
x-geocode	xs:string	*Key Bridge implementation* An enumerated value indicating whether any one of this location object's components have been calculated according to another of this location object's set parameters. *Key Bridge implementation* Allowed values are: • NO (xs:string) (DEFAULT) The *coordinate, address* and *geometry* elements of this *location* are not correlated. • GC (xs:string) The *coordinate.[longitude, latitude]* and *geometry.point* values have been calculated and set according to a Geo-coding algorithm from the address. • RG (xs:string) The *address* has been calculated and set according to a Reverse Geo-coding algorithm from the *coordinate.[longitude, latitude]* value.
x-haat	xs:float	*Key Bridge implementation* The ground height above average terrain value at this location’s coordinates, calculated according to the methodology described in 47 CFR 73.684(d). The elevation model used in the calculation of this *location* attribute is recorded in the *coordinate.x-elevationModel* attribute.

		This value is used to support TVBD transmit antenna compliance with 15.709(b)(2), which states that the ground level HAAT must be below 76 meters.
x-timeZone	xs:string	*Key Bridge implementation* The local time zone at this location. Two interchangeable formats are supported, with the zoneinfo format preferred:[15] • The zoneinfo database format (e.g. "America/New_York") • An offset to Coordinated Universal Time (e.g. "UTC-05:00" or "UTC-5" Note: Three-character notation (e.g. "EDT") is not supported.

8.9.4 Validation

Matching locations: Locations are considered to be equal through analysis of the *location.coordinate.longitude* and *latitude* attributes, which must be equal to within 7 significant digits of each other. For example: one *location* is considered to be equal to another *location* if their *coordinate* values are within 0.00001 degrees of each other (within approximately one (1) meter.) Location equality may be calculated as:

$$Equal = TRUE\ if\ |lat_1 - lat_2| < 0.00001\ AND\ |lon_1 - lon_2| < 0.00001$$

The *x-timeZone* attribute SHOULD use the zoninfo database format wherever possible.

Fully qualified time zone naming notation is required: three-character colloquial notation (e.g. "EDT") is not supported.

[15] See *tz database* online at http://en.wikipedia.org/wiki/Zoneinfo

8.10 preferences

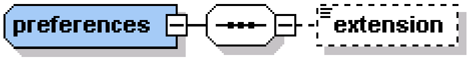


Figure 13: A preferences complex type

WSIF *preferences* allow a user or organization to inform the administrator of default implementation preferences and configurations.

Key Bridge implementation

Key Bridge extends basic preferences to enable the communication and exchange of select user-profile information such as a list of saved schedules, delegated contacts, and locations.

8.10.1 Definition

```
<xs:complexType name="preferences">
        <xs:sequence>
            <xs:element name="extension"            type="xs:string" minOccurs="0"/>
        </xs:sequence>
</xs:complexType>
```

8.10.2 Elements

Element	Type	Documentation
extension	xs:string	A URL-ENCODED string containing key/value pairs to provide additional information or to extend this object.

8.10.3 Attributes

The WSIF *preference* object has no attributes.

8.11 radiationPattern

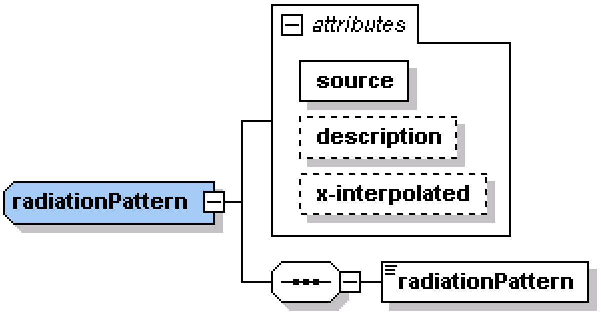


Figure 14: A radiationPattern complex type

The WSIF *radiationPattern* object describes the azimuthal directional gain of a WSIF *antenna*.

The *radiationPattern* element is encoded as simple feature access (SFA) well-known text (WKT).[16]

Notes for US operation:

- All antenna objects built from FCC databases use radiation pattern values submitted by FCC registrants using electronic form 301 / III-B / Tech Box 12.4.

8.11.1 Definition

```
<xs:complexType name="radiationPattern">
        <xs:sequence>
            <xs:element name="radiationPattern"    type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="source"                type="xs:string" use="required"/>
        <xs:attribute name="description"           type="xs:string"/>
        <xs:attribute name="x-interpolated"        type="xs:boolean"/>
    </xs:complexType>
```

[16] See Open Geospatial Consortium Inc., *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture Version 1.2.1*, ed. John R. Herring

8.11.2 Elements

Element	Type	Documentation
radiationPattern	xs:string	The *radiationPattern* represents the directional (azimuth) dependence of the strength of the radio signal from the antenna. A WKT MULTIPOINT SFA Geometry implementation. The azimuthal field values are encoded as a well known text (WKT) MULTIPOINT object with [azimuth, radial value] pairs encoded according to the format POINT(x,y) = POINT(azimuth, field_value). *Key Bridge implementation:* For all directional radiation patterns with incomplete (i.e. greater than one degree of azimuthal sampling) Key Bridge interpolates the missing values to provide guaranteed 360 radial values. TVBD and other antenna radiation patterns may be manually encoded by users, submitted in NSMA-89 file format, or any other data format that an administrator may chose to support.[17]

8.11.3 Attributes

Attribute	Type	Documentation
source	xs:string	The originating source of the data represented in the *radiationPattern* element of this object. An example value for this attribute is "us.fcc.cdbs".
description	xs:string	A free-text description provided for convenience.
x-interpolated	xs:boolean	*Key Bridge implementation* An indicator for whether the *radiationPattern* element of this object contains interpolated values.

8.11.4 Validation

Within the *radiationPattern* element:

- *azimuth* values are measured in degrees and MUST range from 0 to 360 in whole number increments (e.g. 0, 1, 2, 3 … and not 0, 1.5, 2.2 ….)
- antenna receive gain values MUST be normalized and range between 0 and 1

[17] See *Standard Format for Electronic Transfer of Terrestrial Antenna Pattern Data*, National Spectrum Managers Association Report WG 16.89.003 at http://www.nsma.org/recommendation/wg16-89-03.pdf and Federal Communications Commission OET *Interpretation of the NSMA Standard* at http://transition.fcc.gov/oet/info/software/nsma/nsma-intrp.html

8.12 schedule

The WSIF *schedule* object is designed to enable the recording, persistence and distribution of standardized iCalendar-compatible messages.[18]

WSIF *schedule* attributes comprise the minimum set necessary to record and recreate a fully qualified iCalendar event that maintains Rules compliant scheduling information while also retaining marshalling and unmarshalling compatibility with most scheduling software packages including iCal® from Apple® and Outlook® from Microsoft®.[19, 20]

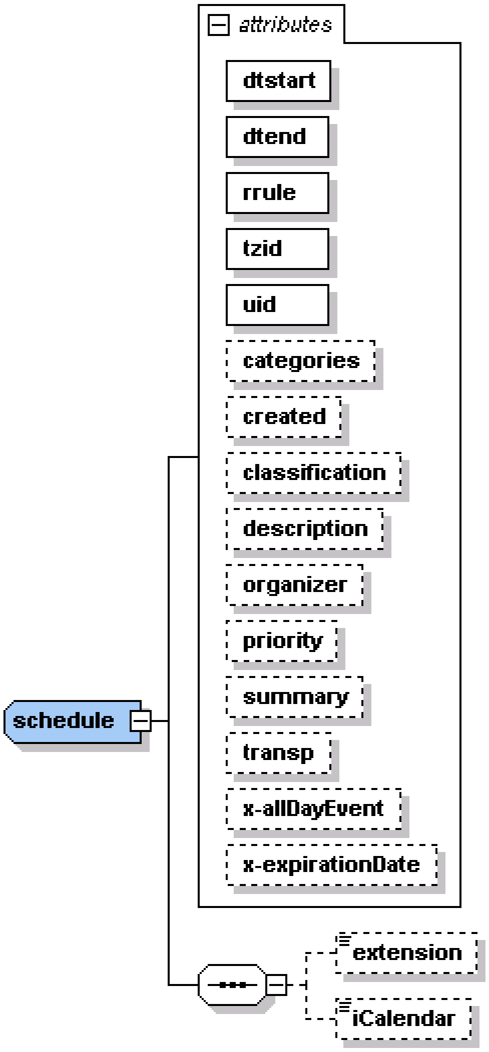


Figure 15: A schedule complex type

[18] See *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, Copyright (c) 2009 IETF Trust and B. Desruisseaux, Oracle at http://tools.ietf.org/html/rfc5545. The attribute definitions of this section are taken from RFC 5545, are Copyright © the IETF Trust, and are reproduced here in accordance with the IETF Trust Legal Provisions Relating to IETF Documents, available online at http://trustee.ietf.org/license-info/IETF-Trust-License-Policy-20091228.htm.

[19] See 47 CFR 15.713(h)(8) and (9), generally "*Wireless Microphones*." Also see 47 CFR 15.713(h)(2) and 74.24, *Short term operation*:

[20] Products are registered trademarks are property of the respective companies.

8.12.1 Definition

```
<xs:complexType name="schedule">
        <xs:sequence>
            <xs:element name="extension" minOccurs="0"/>
            <xs:element name="iCalendar" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="dtstart"             type="xs:dateTime" use="required"/>
        <xs:attribute name="dtend"               type="xs:dateTime" use="required"/>
        <xs:attribute name="rrule"               type="xs:string" use="required"/>
        <xs:attribute name="tzid"                type="xs:string" use="required"/>
        <xs:attribute name="uid"                 type="xs:string" use="required"/>
        <xs:attribute name="categories"          type="xs:string"/>
        <xs:attribute name="created"             type="xs:dateTime"/>
        <xs:attribute name="classification"      type="xs:string"/>
        <xs:attribute name="description"         type="xs:string"/>
        <xs:attribute name="organizer"           type="xs:string"/>
        <xs:attribute name="priority"            type="xs:int"/>
        <xs:attribute name="summary"             type="xs:string"/>
        <xs:attribute name="transp"              type="xs:string"/>
        <xs:attribute name="x-allDayEvent"       type="xs:boolean"/>
        <xs:attribute name="x-expirationDate"    type="xs:dateTime"/>
</xs:complexType>
```

8.12.2 Elements

Element	Type	Documentation
extension	xs:string	A URL-ENCODED string containing key/value pairs that provides additional information or extends this object.
iCalendar	xs:string	This WSIF *schedule* object includes only a subset of all possible values that may be provided from a RFC 5545-compliant iCalendar file. If this *contact* was created from an iCalendar file then the original iCalendar may be stored for convenience and future reference.[21]

[21] See IETF RFC 5545: *Internet Calendaring and Scheduling Core Object Specification (iCalendar)* at http://tools.ietf.org/html/rfc5545

8.12.3 Attributes

The following attribute descriptions are excerpted from RFC 5545, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)* available online at http://tools.ietf.org/html/rfc5545.

Attribute	Type	Documentation (from RFC 5545)
dtstart	xs:dateTime	3.8.2.4. Date-Time Start The "DTSTART" property for a "VEVENT" specifies the inclusive start of the event. For recurring events, it also specifies the very first instance in the recurrence set. *Key Bridge implementation* DTSTART is formatted according to the DATE-TIME definition for "floating" values and are not bound to any time zone in particular.[22]
dtend	xs:dateTime	3.8.2.2. Date-Time End The "DTEND" property for a "VEVENT" calendar component specifies the non-inclusive end of the event. For cases where a "VEVENT" calendar component specifies a "DTSTART" property with a DATE value type but no "DTEND" nor "DURATION" property, the event's duration is taken to be one day. *Key Bridge implementation* DTEND is formatted according to the DATE-TIME definition for "floating" values and are not bound to any time zone in particular.[23]
rrule	xs:string	Implementation note: WSIF recurrence differs slightly from the RFC specification. Specifically: • RDATE AND EXDATE ATTRIBUTES ARE NOT USED The WSIF *rrule* attribute definition is provided below with modifications noted by strike-through: 3.8.5.3. Recurrence Rule This *rrule* attribute defines a rule or repeating pattern for recurring events or time zone definitions. The recurrence rule, if specified, is used in computing the recurrence set. The recurrence set is the complete set of recurrence instances for a calendar component. The recurrence set is generated by considering the initial "DTSTART" property along with the "RRULE"~~, "RDATE", and "EXDATE"~~ properties contained within the recurring component. The "DTSTART" property defines the first instance in the recurrence set. The "DTSTART" property value SHOULD be synchronized with the recurrence rule, if specified. The recurrence set generated with a "DTSTART" property value not synchronized with the recurrence rule is undefined. The final recurrence set is generated by gathering all of the start DATE-TIME values generated by any of the specified "RRULE"~~ and "RDATE"~~ properties~~, and then excluding any start~~

[22] See Section 3.3.5. *Date-Time* in RFC 5545 at http://tools.ietf.org/html/rfc5545#section-3.3.5

[23] Ibid.

		~~DATE-TIME values specified by "EXDATE" properties~~.
tzid	xs:string	3.2.19. Time Zone Identifier [Used] to specify the identifier for the time zone definition for a time component in the property value. Implementers may want to use the naming conventions defined in existing time zone specifications such as the public-domain TZ database [TZDB]. [24] *Key Bridge implementation* The specification of globally unique time zone identifiers follows the formats identified in the public-domain TZ database. For example: America/New_York
uid	xs:string	3.8.4.7. Unique Identifier This property defines the persistent, globally unique identifier for the calendar component.
categories	xs:string	3.8.1.2. Categories This property defines the categories for a calendar component. Some possible English values for "CATEGORIES" property include: "ANNIVERSARY", "APPOINTMENT", "BUSINESS", "EDUCATION", "HOLIDAY", "MEETING", "MISCELLANEOUS", "NON-WORKING HOURS", "NOT IN OFFICE", "PERSONAL", "PHONE CALL", "SICK DAY", "SPECIAL OCCASION", "TRAVEL", and " VACATION". Categories can be specified in any registered language. *Key Bridge implementation* All *schedule* objects are set by default to a "WHITESPACE" category plus any additional categories provided by users.
created	xs:dateTime	3.8.7.1. Date-Time Created This property specifies the date and time that the calendar information was created by the calendar user agent in the calendar store.
classification	xs:string	3.8.1.3. Classification This property corresponds to the iCalendar property "CLASS" which defines the access classification for a calendar component. Examples include "PUBLIC", "PRIVATE", and "CONFIDENTIAL".
description	xs:string	3.8.1.5. Description This property provides a more complete description of the calendar component than that provided by the "SUMMARY" property, [and is used in the "VEVENT" to] capture lengthy textual descriptions associated with the activity
duration	xs:string	3.3.6. Duration and 3.8.2.5. Duration This property specifies a positive duration of time.

[24] Eggert, P. and A.D. Olson, "*Sources for Time Zone and Daylight Saving Time Data*", July 2009, http://www.twinsun.com/tz/tz-link.htm

		In a "VEVENT" calendar component the property may be used to specify a duration of the event, instead of an explicit end DATE-TIME. The format is based on the ISO.8601.2004 complete representation basic format with designators for the duration of time.[25] The format can represent nominal durations (weeks and days) and accurate durations (hours~~, minutes, and seconds~~). Note that unlike ISO.8601.2004, this value type doesn't support the "Y" and "M" designators to specify durations in terms of years and months Example: The following is an example of this property that specifies an interval of time of one hour and zero minutes and zero seconds: DURATION:PT1H0M0S
organizer	xs:string	3.8.4.3. Organizer This property defines the organizer for a calendar component. This property MUST be specified in an iCalendar object that specifies a group-scheduled calendar entity. This property MUST be specified in an iCalendar object that specifies the publication of a calendar user's busy time. This property MUST NOT be specified in an iCalendar object that specifies only a time zone definition or that defines calendar components that are not group-scheduled components, but are components only on a single user's calendar. The following is an example of this property: ORGANIZER;CN=John Smith:mailto:jsmith@example.com
priority	xs:int	3.8.1.9. Priority This property defines the relative priority for a calendar component. This priority is specified as an integer in the range 0 to 9. A value of 0 specifies an undefined priority. A value of 1 is the highest priority. A value of 2 is the second highest priority. Subsequent numbers specify a decreasing ordinal priority. A value of 9 is the lowest priority.
summary	xs:string	3.8.1.12. Summary This property defines a short summary or subject for the calendar component. It contains the text to be used as the message subject
transp	xs:string	3.8.2.7. Time Transparency This property defines whether or not an event is transparent to busy time searches. Time Transparency is the characteristic of an event that determines whether it appears to consume time on a calendar. Events that consume actual time for the individual or resource associated with the calendar SHOULD be recorded as OPAQUE, allowing them to be detected by free/busy time searches. Other events, which do not take up the individual's (or resource's) time SHOULD be recorded as TRANSPARENT, making them invisible to free/busy time searches.

[25] See International Organization for Standardization, "*Data elements and interchange formats -- Information interchange -- Representation of dates and times*", 2004.

		For example: • TRANSP:TRANSPARENT Describes an event that is transparent or does not block on free/busy time searches: • TRANSP:OPAQUE Describes an event that is opaque or blocks on free/busy time searches:
extension		
iCalendar		
x-allDayEvent	xs:boolean	*Key Bridge implementation* Indicates whether this event is an all-day event (e.g. if set to 'TRUE' the event runs for 24 hours).
x-expirationDate	xs:dateTime	*Key Bridge implementation* The expiration date of this WSIF *schedule* as calculate from the start and end dates and this *schedule*'s recurrence configuration. For example: a one-day event (e.g. *dtstart* equal to *dtend*) with a recurrence for ten (10) days will have an *expiration* attribute set to ten (10) calendar days after DTSTART.

8.12.4 Validation

If more than one category is indicated they must be formatted as comma-delimited text with no spaces. For example: "CATEGORIES:WHITESPACE,LICENSED,LPTV".

The RDATE and EXDATE parameters are not supported by WSIF.

Within a WSIF-defined iCalendar event the following constraints are placed on the recurrence rule (*rrule*) attribute:[26]

- Valid FREQ values are: HOURLY, DAILY, WEEKLY, MONTHLY, YEARLY
- SECONDLY and MINUTELY FREQ values are not supported
- UNTIL must be less than or equal to 1 year from the *created* attribute and should be automatically corrected if invalid
- COUNT values that would extend recurrence beyond one (1) year from the *created* attribute should be automatically corrected to limit event duration to a maximum of one (1) year.
- If no end date (UNTIL) is provided, one should be automatically created as equal to one (1) day after from the *created* attribute
- BYSECOND and BYMINUTE rule parts are not supported
- The week start parameter WKST is defined as always being Monday ("MO")
- BYSETPOS is limited to a maximum of one single integer value. It is an ERROR to provide more than one value.
- BYSETPOS may only be used with a MONTHLY recurrence frequency.
  i.e. "The first Sunday of every 2nd week" IS supported (MONTHLY), whereas "The third hour of every second day" (DAILY) or "The last work day or every week" (WEEKLY) or "The second month of every year" (YEARLY) are NOT supported.

If the *x-allDayEvent* attribute is set to "TRUE" then the *schedule* MUST be interpreted to begin and end exactly at midnight on the respective indicated start and end dates.

From RFC 5545 Section 3.8.5.3. *Recurrence Rule*

- If the *duration* of the given recurrence instance is modified, then all subsequence instances are also modified to have this same *duration*.
- If the *duration* of the recurring component is specified with the "DTEND" ~~or "DUE"~~ property, then the same exact *duration* will apply to all the members of the generated recurrence set. Else, if the *duration* of the recurring component is specified with the "DURATION" property, then the same nominal *duration* will apply to all the members of the generated recurrence set and the exact *duration* of each recurrence instance will depend on its specific start time.

From RFC 5545 Section 5. *Recommended Practices*:

- An implementation can truncate a "SUMMARY" property value to 255 octets, but it MUST NOT truncate the value in the middle of a UTF-8 multi-octet sequence.
- If seconds of the minute are not supported by an implementation, then a value of "00" SHOULD be specified for the seconds component in a time value.

[26] See RFC 5545 Section 3.3.10. *Recurrence Rule* at http://tools.ietf.org/html/rfc5545#section-3.3.10

8.13 securityParameter

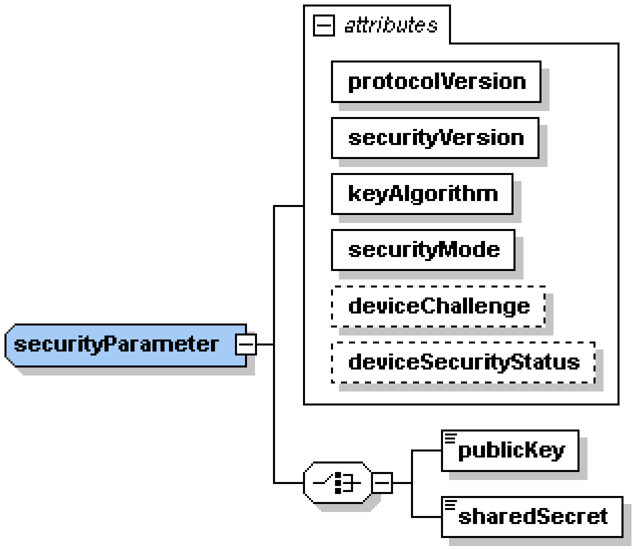


Figure 16: A securityParameter complex type

The WSIF *securityParameter* object provides a standardized container to hold and distribute security-related parameters.

The *securityParameter* object is designed to support both human and machine-to-machine authentication and includes accommodation for public keys and shared secrets plus necessary information to enable successful machine-to-machine credential exchange and handshaking.

The WSIF *securityParameter* object is not intended to be used atomically, but rather is intended to be incorporated into, and thereby add security credentialing, to other WSIF objects.

8.13.1 Definition

```
<xs:complexType name="securityParameter">
        <xs:choice>
            <xs:element name="publicKey"          type="xs:base64Binary"/>
            <xs:element name="sharedSecret"       type="xs:base64Binary"/>
        </xs:choice>
        <xs:attribute name="protocolVersion"      type="xs:float" use="required"/>
        <xs:attribute name="securityVersion"      type="xs:float" use="required"/>
        <xs:attribute name="keyAlgorithm"         type="xs:string" use="required"/>
        <xs:attribute name="securityMode"         type="xs:string" use="required"/>
        <xs:attribute name="deviceChallenge"      type="xs:string"/>
        <xs:attribute name="deviceSecurityStatus" type="xs:string"/>
    </xs:complexType>
```

8.13.2 Elements

Element	Type	Documentation
publicKey	xs:base64Binary	If a public key cryptography is used, the counter-party's public key may be store here.
sharedSecret	xs:base64Binary	If shared secret cryptography is used, the shared secret passphrase may be stored here.

8.13.3 Attributes

Attribute	Type	Documentation
protocolVersion	xs:float	The version of this security information container.
securityVersion	xs:float	The version of this security information container.
keyAlgorithm	xs:string	The cryptographic algorithm used to create the public/private key pair or shared secred. *Key Bridge implementation* Allowed values are: • DSA (Digital Signature Algorithm) • RSA DSA is specified by FIPS 186-3 and must be exactly 1024 bits. DSA keys are preferred and shall be employed by default. For RSA keys, the minimum size is 768 bits and the default is 2048 bits. Generally, 2048 bits is considered sufficient.
securityMode	xs:string	Indicates the security method. *Key Bridge implementation* Allowed values are: • PUBLICKEY • SHAREDSECRET Additional values may be extended in a future version.
deviceSecurityStatus	xs:string	In a machine-to-machine transaction, this element messages the device's current security status. NOTE: Implementation specifics may be determined according to user requirements. Example status messages may include: • KEY_VALIDATED • KEY_NOT_VALIDATED • NEW_KEY_ASSIGNED, etc.
deviceChallenge	xs:string	In a machine-to-machine transaction, this element identifies additional information requested from the client. Examples many include a nonce (number used once) value for initiating digest access authentication.

8.14 station

A WSIF *station* describes the physical location of a transmitting or receiving wireless service. For example, a WSIF *station* could describe the current location of a mobile-phone user, a satellite uplink facility, a wireless microphone transmitter, a TVBD user, etc.

The WSIF *station* is intended to be a generalized object template that may support many variants of wireless facility and location where wireless operations may occur. Station object validity rules are therefore determined by each wireless service implementation and may be defined through the *stationClass* and *stationType* attributes.

Implementation Note:

All transmitting stations (and specifically white space devices) must provide certain transmission parameters (e.g. FCC ID and serial number). These parameters are set by adding one or more WSIF *transmitter* objects to the *transmitterList* element.

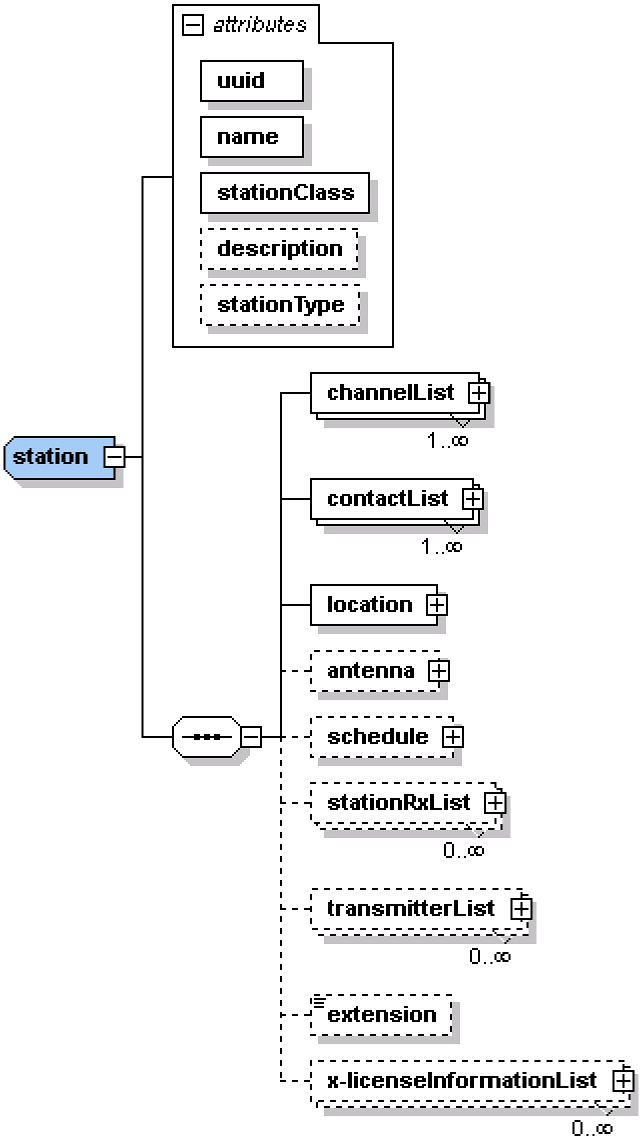


Figure 17: A station complex type

8.14.1 Definition

```
<xs:complexType name="station">
        <xs:sequence>
            <xs:element name="channelList"          type="channel" maxOccurs="unbounded"/>
            <xs:element name="contactList"          type="contact" maxOccurs="unbounded"/>
            <xs:element name="location"             type="location"/>
            <xs:element name="antenna"              type="antenna" minOccurs="0"/>
            <xs:element name="schedule"             type="schedule" minOccurs="0"/>
            <xs:element name="stationRxList"        type="station" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="transmitterList"      type="transmitter" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="extension"            type="xs:string" minOccurs="0"/>
            <xs:element name="x-licenseInformationList"   type="licenseInformation"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="uuid"               type="xs:string" use="required"/>
        <xs:attribute name="name"               type="xs:string" use="required"/>
        <xs:attribute name="stationClass"       type="xs:string" use="required"/>
        <xs:attribute name="description"        type="xs:string"/>
        <xs:attribute name="stationType"        type="xs:string"/>
    </xs:complexType>
```

8.14.2 Elements

Element	Type	Documentation
channelList	channel	A list of channels (i.e. frequency ranges) that are occupied by the transmitter(s) at this station
contactList	contact	A list of contacts associated with this station. For example, a facility or on-site technical manager, administrator, and operational contacts may be identified. *Key Bridge implementation* A minimum of one WSIF *contact* is required by default but may be declared optional for certain WSIF *station* types (e.g. for personal/portable TV Band devices). If this WSIF *station* is incorporated as a component within a *wirelessService* object then one or more of this station's *contact* elements may refer to the parent *wirelessService contact* element by XPATH location.
location	location	Information describing this station's physical and geographic location. Station
antenna	*antenna*	A description of this station's (transmit or receive) antenna.

schedule	schedule	The transmitting schedule of this station.
stationRxList	*station*	For wireless services that include multiple stations, and especially for wireless services with multiple TXRX stations, each receiving station may indicate its respective upstream transmitting stations by adding that transmitting *station* to this receiving station's *rxStationList* element. Example uses of this element include Television translator stations, MVPD receive sites, etc.
transmitterList	transmitter	This WSIF *station* may support multiple transmitters operating within the same geographic area and on the same schedule. For example, several wireless microphones may operate simultaneously within the geographic contour defined within this station's *location* element. If the *stationType* attribute indicates this WSIF *station* is receive-only then this element SHOULD be null.
extension	xs:string	A URL-ENCODED string containing key/value pairs to provide additional information or to extend this object.
x-licenseInformationList	licenseInformation	*Key Bridge implementation* If the wireless service is licensed to or operates under to one or more call signs, then each respective license information element may be associated to this station by adding it to this list.

8.14.3 Attributes

Attribute	Type	Documentation
uuid	xs:string	A universally unique identifier (UUID) associated with and permanently assigned to this *station*. *Key Bridge implementation* See the attribute description for *location.uuid* for an explanation of how stations are matched according to their location.
name	xs:string	A human-readable name or label of this station. *Key Bridge implementation* For licensed wireless services it is recommended to use the facility call sign. For unlicensed wireless services it is recommended to use the venue name.
stationClass	xs:string	Indicates the station classification. Classification may be used to determine whether and how many elements of this *station* are required for validation. *Key Bridge implementation* Allowed values are: • TX This is a transmitting station and a *transmitter* is required in the *transmitterList* element • RX This is a receive-only station and a *transmitter* element is NOT required • TXRX This station is able to both transmit and receive and a *transmitter* is required in the *transmitterList* element
description	xs:string	A human-readable description of this *station*, which is useful for providing more detail about a station and to supplement the name attribute. *Key Bridge implementation* This *description* attribute is set to a human-readable translation of this station's *stationClass* attribute.
stationType	xs:string	A description of this station's operation.

8.14.4 Validity

If the *stationClass* attribute is set to "TX" or "TXRX" then at least one *transmitter* is required in the *transmitterList* element.

At least one *channel* is required in the *channelList* element.

A station *location* element must include a *coordinate* and optionally an *address* plus a *geometry.polygon* (e.g. a geographic contour).

The *antenna* element may be set to null if and only if the station employs a unity-gain, omni-directional antenna.

Whenever this WSIF *station* is derived from an FCC database and is licensed under that call sign, a WSIF *licenseInformation* object must be added to the *licenseInformationList* element with the appropriate call sign information.

8.15 transmitter

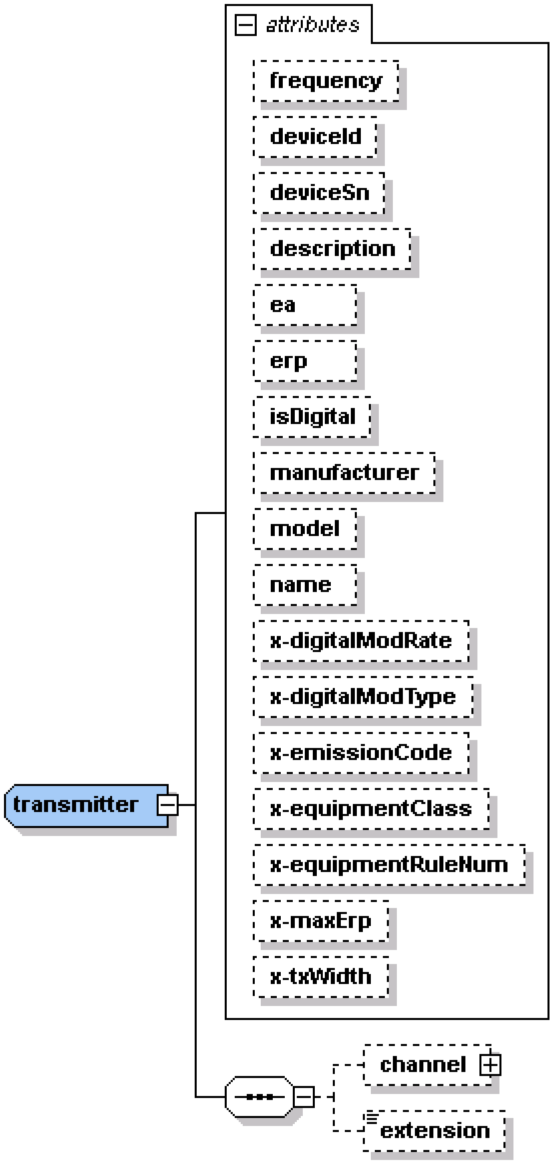


Figure 18: A transmitter complex type

The WSIF *transmitter* object provides an extensible software template to contain and exchange required and optional but otherwise useful transmitter information.

A WSIF *transmitter* provides an object template for common device-related attributes and may be optionally used directly or, more likely, may be extended by other, more specific transmitter descriptions that fully describe a certain type wireless device. For this reason all WSIF transmitter attributes and elements are defined as optional by default. Attribute and element validity is expected to be defined in *transmitter*-derived objects.

The WSIF *transmitter* is designed to support, through extension, the communication of required and optional but otherwise useful information about licensed and unlicensed wireless devices including transmitters, receivers and transceivers.

8.15.1 Definition

```
<xs:complexType name="transmitter">
    <xs:sequence>
      <xs:element name="channel"            type="channel" minOccurs="0"/>
      <xs:element name="extension"          type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="frequency"          type="xs:double"/>
    <xs:attribute name="deviceId"           type="xs:string"/>
    <xs:attribute name="deviceSn"           type="xs:string"/>
    <xs:attribute name="description"        type="xs:string"/>
    <xs:attribute name="ea"                 type="xs:string"/>
    <xs:attribute name="erp"                type="xs:float"/>
    <xs:attribute name="isDigital"          type="xs:boolean"/>
    <xs:attribute name="manufacturer"       type="xs:string"/>
    <xs:attribute name="model"              type="xs:string"/>
    <xs:attribute name="name"               type="xs:string"/>
    <xs:attribute name="x-digitalModRate"   type="xs:double"/>
    <xs:attribute name="x-digitalModType"   type="xs:string"/>
    <xs:attribute name="x-emissionCode"     type="xs:string"/>
    <xs:attribute name="x-equipmentClass"   type="xs:string"/>
    <xs:attribute name="x-equipmentRuleNum" type="xs:string"/>
    <xs:attribute name="x-maxErp"           type="xs:float"/>
    <xs:attribute name="x-txWidth"          type="xs:float"/>
  </xs:complexType>
```

8.15.2 Elements

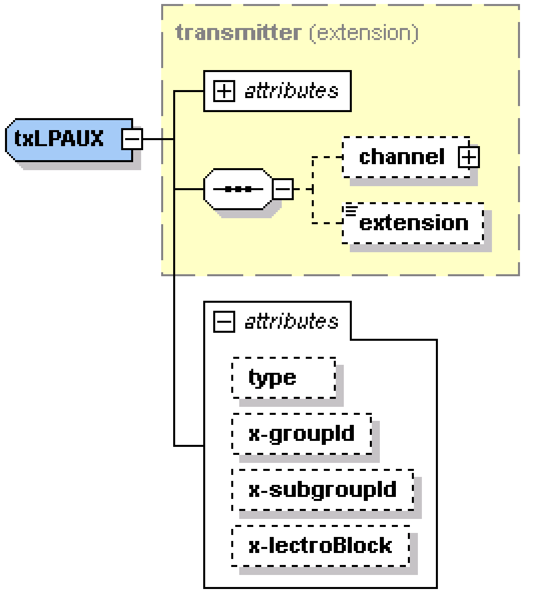
Element	Type	Documentation
channel	channel	This transmitter’s operating channel
extension	xs:string	A URL-ENCODED string containing key/value pairs that administrators may implement at their discretion to provide additional information or to extend this object.

8.15.3 Attributes

Attribute	Type	Documentation
frequency	xs:double	If a specific frequency has been assigned to this transmitter that information may be recorded here in MHz *Key Bridge implementation* If only the channel is provided then the *assignedFrequency* value is set to the center frequency of this transmitter’s *channel*.
deviceId	xs:string	The transmitter device’s Government-provided identifier.
deviceSn	xs:string	The transmitting device’s manufacturer-provided serial number.

description	xs:string	A human-readable description of the transmitting device.
ea	xs:string	The Government equipment authorization agency from which this device is authorized to operate and which issued the device ID. *Key Bridge implementation* Agencies are identified in reverse dot delimited notation. See *Appendix: Enumerated Codes, Equipment authorization agencies*.
erp	xs:float	The transmitting device's current effective radiated power (ERP), measured in dBw.
isDigital	xs:boolean	Indicates whether this transmitter is sending a digital (TRUE) or analog (FALSE) carrier
manufacturer	xs:string	The device manufacturer (company name)
model	xs:string	The device product model
name	xs:string	A user-defined, convenience name used to identify the device.
x-digitalModRate	xs:float	*Key Bridge implementation* This *transmitters'* wireless digital modulation information rate, measured in bits per second (bps)
x-digitalModType	xs:string	*Key Bridge implementation* This *transmitter's* wireless digital modulation type or scheme. For example: "QPSK", "VSB", "QAM", etc.
x-emissionCode	xs:string	*Key Bridge implementation* This *transmitter's* wireless emission designator. This value is typically associated with licensed services, and is used to encode the type of modulation of the main carrier, the nature of the modulating signals, and the type of information to be transmitted.
x-equipmentClass	xs:string	*Key Bridge implementation* An enumerated value that identifies the equipment classification according to its authorized functions and Rule parts. NOTE: The *x-equipmentClass* attribute and *x-equipmentRuleNum* attributes may be redundant when describing equipment authorized by the FCC.
x-equipmentRuleNum	xs:string	*Key Bridge implementation* The regulatory Rule under which this device is authorized to operate.
x-maxERP	xs:float	*Key Bridge implementation* The maximum transmitting level (in dBW) of this *transmitter*.
x-txWidth	xs:float	*Key Bridge implementation* The actual channel width of this *transmitter's* carrier signal.

8.16 txLPAUX



The WSIF *txLPAUX* object extends the general WSIF *transmitter* to add certain attributes that are specific to and useful for the management of wireless microphones.

Figure 19: A txLPAUX complex type, which extends the transmitter complexType

8.16.1 Definition

```
<xs:complexType name="txLPAUX">
        <xs:complexContent>
            <xs:extension base="transmitter">
                <xs:attribute name="type"          type="xs:string"/>
                <xs:attribute name="x-groupId"     type="xs:string"/>
                <xs:attribute name="x-subGroupId"  type="xs:string"/>
                <xs:attribute name="x-lectroBlock" type="xs:int"/>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>
```

8.16.2 Elements

The WSIF *txLPAUX* object adds no new elements to the *transmitter* interface.

8.16.3 Attributes

txLPAUX extends the transmitter interface and adds the following unique attributes.

Attribute	Type	Documentation
type	xs:string	The type of LPAUX device. e.g. “Wireless microphone”, “IFB”, etc.
x-groupId	xs:string	*Key Bridge implementation* A convenience parameter that may be used to organize or categorize a group of wireless devices.
x-subGroupId	xs:string	*Key Bridge implementation* Convenience parameters that may be used to organize or categorize a group of wireless devices under a top-level organization established by the *groupId* attribute.
x-lectroBlock	xs:int	*Key Bridge implementation* Operating frequency block number specific to Lectrosonics brand microphones.

8.16.4 Validation

The transmitter *channel* element is required.

8.17 txTVBD

The WSIF *txTVBD* object extends the general WSIF *transmitter* to add certain attributes that are specific to, useful for, and required to implement the management of unlicensed TV Band devices.

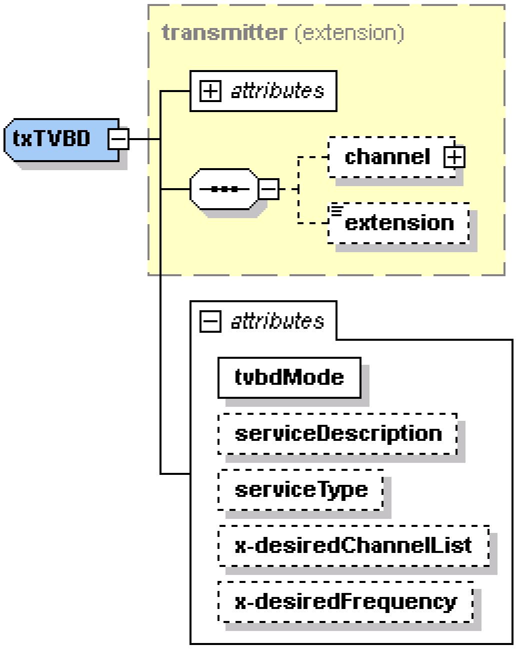


Figure 20: : A txTVBD complex type, which extends the transmitter complexType

8.17.1 Definition

```
<xs:complexType name="txTVBD">
        <xs:complexContent>
            <xs:extension base="transmitter">
                <xs:attribute name="tvbdMode"      type="xs:string" use="required"/>
                <xs:attribute name="serviceDescription"      type="xs:string"/>
                <xs:attribute name="serviceType"      type="xs:string"/>
                <xs:attribute name="x-desiredChannelList"  type="xs:string"/>
                <xs:attribute name="x-desiredFrequency"     type="xs:float"/>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>
```

8.17.2 Elements

The WSIF *txTVBD* object adds no new elements to the *transmitter* interface.

8.17.3 Attributes

txTVBD extends the transmitter interface and adds the following unique attributes.

Attribute	Type	Documentation
tvbdMode	xs:string	The TVBD operating mode. *Key Bridge implementation* Allowed values are: • FIXED The TVBD is a fixed type white space device • TRANSPORTABLE The TVBD is a Mode II type white space device operating in a non-mobile (i.e. stationary) configuration • MOBILE The TVBD is a Mode II type white space device in a mobile (i.e. actively in motion) configuration • CLIENT The TVBD is operating as a Mode I type white space device Note: The CLIENT operating mode is included for completeness. Additional values may be added in the future as requirements demand.
desiredChannelList	xs:string	A URL-ENCODED string containing key/value pairs that identify a list of channels that the TVBD prefers to operate on. *Key Bridge implementation* Allowed keys are: • allocation A reverse dot-delimited description of the frequency allocation. See the *channel.allocation* attribute for additional details. • desiredChannelList A comma delimited list of TV channels identified as a whole integer (xs:int) Additional key-value pairs may be added as necessary. An example *desiredChannelList* attribute value is shown below: allocation=us.fcc.broadcast.174-216&desiredChannelList=7,8,9
desiredFrequency	xs:float	A specific frequency (in MHz) that the device wishes to inquire about or operate on.
serviceDescription	xs:string	A human-readable description of the radio service the device may optionally provide. (e.g. WiFi 802.11 xyz).
serviceType	xs:string	A machine-readable, encoded identifier of the radio service the requesting device wishes to provide. *Key Bridge implementation* Please refer to *Appendix: Wireless service type* for an enumerated list of named values and their description.

8.17.4 Validation

The *txTVBD* attribute *tvbdMode* is required.

The *transmitter* attribute *deviceId* is required and must be set to the device's FCC ID.

The *transmitter* attribute *deviceSn* is required and must be set to the device's manufacturer serial number.

The *transmitter* attribute *ea* is required.

Key Bridge implementation

For US white space operation the transmitter attribute *ea* is set to "us.fcc".

8.18 txTV

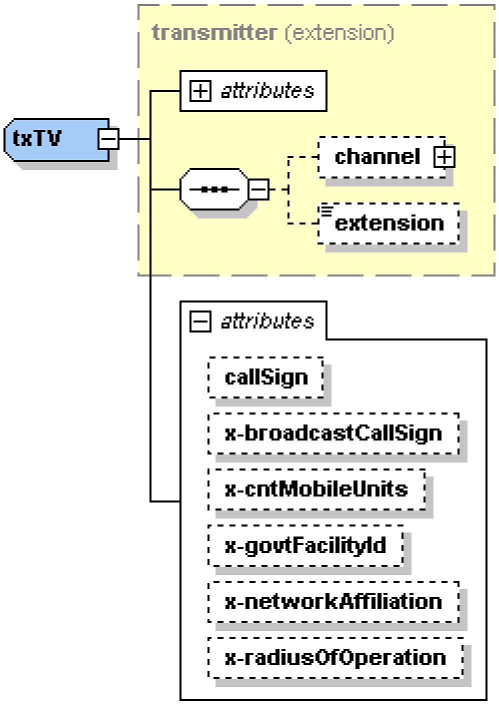


Figure 21: : A txTV complex type, which extends the transmitter complexType

Broadcast Television transmitters may be described using the *txTV* object, which extends the *transmitter* interface to include a number of device attributes required for the description of broadcast television services.

8.18.1 Definition

```
<xs:complexType name="txTV">
        <xs:complexContent>
            <xs:extension base="transmitter">
                <xs:attribute name="callSign"           type="xs:string"/>
                <xs:attribute name="x-broadcastCallSign"    type="xs:string"/>
                <xs:attribute name="x-cntMobileUnits"       type="xs:int"/>
                <xs:attribute name="x-govtFacilityId"       type="xs:string"/>
                <xs:attribute name="x-networkAffiliation"   type="xs:string"/>
                <xs:attribute name="x-radiusOfOperation"    type="xs:int"/>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>
```

8.18.2 Elements

The WSIF *txTV* object adds no new elements to the *transmitter* interface.

8.18.3 Attributes

txLPAUX extends the transmitter interface and adds the following unique attributes.

Attribute	Type	Documentation
callSign	xs:string	If this *transmitter* supports a licensed wireless service, that wireless service's call sign should be recorded. *Key Bridge implementation* Note: This *callSign* attribute may be duplicative of a value in the *wirelessService.licenseInformationList* element.
x-broadcastCallSign	xs:string	*Key Bridge implementation* If this *transmitter* is associated with a licensed broadcast television service, that service's broadcast call sign may be recorded for convenience.
x-cntMobileUnits	xs:int	*Key Bridge implementation* The number of client devices this station supports.
x-govtFacilityId	xs:int	*Key Bridge implementation* If this *transmitter* is derived from a Government database record, that record's facility or location ID (e.g. a CDBS *facility_id* or ULS *location* number) may be recorded for convenience and for cross-referencing.
x-networkAffiliation	xs:string	*Key Bridge implementation* If this *transmitter* is associated with a licensed broadcast television service, that service's network affiliation may be recorded for convenience.
x-radiusOfOperation	xs:double	*Key Bridge implementation* If this *transmitter* supports mobile clients or is itself mobile then the typical or licensed radius of operation may be recorded for reference. Radius of operation is recorded in kilometers (km) and is centered upon the *location.coordinate* of this *transmitter's* associated *station*.

8.18.4 Validation

The *transmitter* element *channel* is required.

The *transmitter* attribute *ea* is required.

The *transmitter* attribute *erp* is required.

The *transmitter* attribute *isDigital* is required.

Key Bridge implementation

All transmitters derived from FCC databases have their attribute *ea* set to "us.fcc".

8.19 wirelessService

The WSIF *wirelessService* object is a generalized container for any type of wireless service and accommodates the decoupled description of transmitting and receiving services.

The WSIF *wirelessService* is specifically designed to accommodate all wireless services described within the white space Rules, including licensed and unlicensed services.

Various wireless service implementations may require and contain different elements and attributes, which may be identified according to the *wirelessServiceType* attribute and each incorporate elements own Type attributes, which are intended to describe the represented service in a machine-readable format to enable automated processing and element validation.

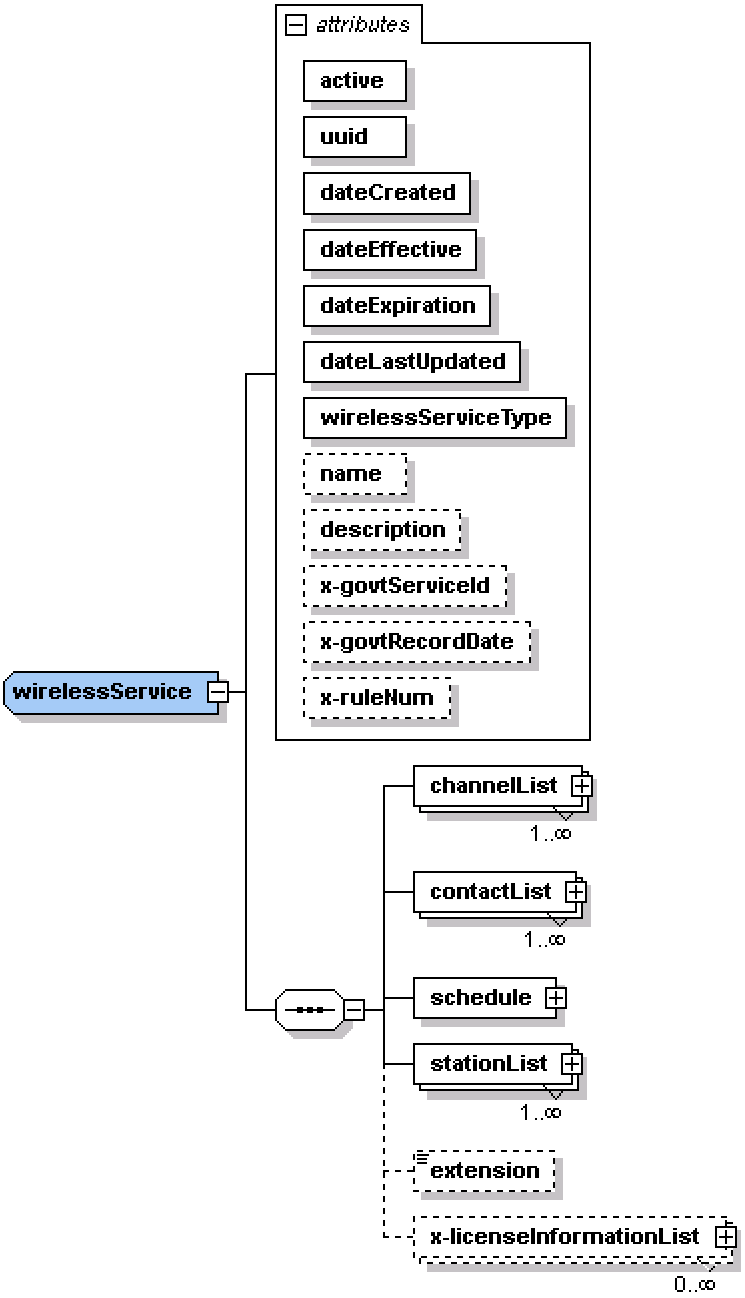


Figure 22: : A wirelessService complex type

8.19.1 Definition

```
<xs:complexType name="wirelessService">
        <xs:sequence>
            <xs:element name="channelList"          type="channel" maxOccurs="unbounded"/>
            <xs:element name="contactList"          type="contact" maxOccurs="unbounded"/>
            <xs:element name="schedule"             type="schedule"/>
            <xs:element name="stationList"          type="station" maxOccurs="unbounded"/>
            <xs:element name="extension"            type="xs:string" minOccurs="0"/>
            <xs:element name="x-licenseInformationList"   type="licenseInformation"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="active"                 type="xs:boolean" use="required"/>
        <xs:attribute name="uuid"                   type="xs:string" use="required"/>
        <xs:attribute name="dateCreated"            type="xs:dateTime" use="required"/>
        <xs:attribute name="dateEffective"          type="xs:dateTime" use="required"/>
        <xs:attribute name="dateExpiration"         type="xs:dateTime" use="required"/>
        <xs:attribute name="dateLastUpdated"        type="xs:dateTime" use="required"/>
        <xs:attribute name="wirelessServiceType"    type="xs:string" use="required"/>
        <xs:attribute name="name"                   type="xs:string"/>
        <xs:attribute name="description"            type="xs:string"/>
        <xs:attribute name="x-govtServiceId"        type="xs:int"/>
        <xs:attribute name="x-govtRecordDate"       type="xs:dateTime"/>
        <xs:attribute name="x-ruleNum"              type="xs:string"/>
</xs:complexType>
```

8.19.2 Elements

Element	Type	Documentation
channelList	channel	A list of channels (i.e. frequency ranges) that are used by this wireless service. *Validation rule* At least one channel is required.
contactList	contact	A list of contacts associated with this wireless service. Contacts are required by default, but may be declared optional for certain wireless service types (e.g. for personal/portable TV Band devices).
schedule	schedule	If the wireless service operates intermittently then the schedule must identify the hours of operation. Station-specific schedules may be used when a wireless service contains more than one station and each station transmits according to its own schedule. In this case, the Wireless Service schedule must be the UNION of all its component stations. For example: if station A transmits from 06:00 to 12:00 and station B transmits from 10:00 to 18:00, the Wireless Service schedule would be 06:00 to 18:00. Schedules for full time operations begin on the *effectiveDate* and end on the *expirationDate* of this *wirelessService*. If the *effectiveDate* and *expirationDate* fall on the same day then the duration must be interpreted as exactly one day, from midnight to midnight.
stationList	station	A list of stations associated with this wireless service. Station types and count may vary according to the *wirelessServiceType* and actual station configurations. *Key Bridge implementation* See implementation examples.
extension	xs:string	A URL-ENCODED string containing key/value pairs that administrators may implement at their discretion to provide additional information or to extend this object. *Key Bridge implementation* May contain the following key/value pairs: • agree (xs:boolean) Rrequired if active lpaux/mvpd/tvbd service. Mapped to the xs:string values "true" and "false" • signature (xs:string) Required if active paux/mvpd/tvbd service • notes (xs:string) User free text

x-licenseInformationList	licenseInformation	*Key Bridge implementation* If the wireless service is licensed to or operates under to one or more call signs, then each respective license information element may be associated to this wireless service by adding it to this list.

8.19.3 Attributes

Attribute	Type	Documentation
active	xs:boolean	Indicates whether this wireless service is active. Allowed values are: true: indicates this wirelessService is active false: indicates this wirelessService is not active The default value is "true". *Key Bridge implementation* Inactive wireless services (active = "false") represent reservations that have been created but not yet activated by their respective owner. Only active wireless services are considered when evaluating and establishing white space protections.
uuid	xs:string	A universally unique identifier (UUID) associated with and permanently assigned to this registered wireless service.
dateCreated	xs:dateTime	The date and time with this wireless service was created by the responsible party
dateEffective	xs:dateTime	The date and time when this wireless service record should be considered effective. Note that the effective date may be set in the future to accommodate advanced registration of wireless services.
dateExpiration	xs:dateTime	The date and time when this wireless service record expires. *Key Bridge implementation* The expiration date of this WSIF *schedule* as calculate from the start and end dates and this *schedule*'s recurrence configuration. For example: a one-day event (e.g. *dtstart* equal to *dtend*) with a recurrence for ten (10) days will have an *expiration* attribute set to ten (10) calendar days after DTSTART.
dateLastUpdated	xs:dateTime	The date and time this wireless service was last updated by the responsible party
wirelessServiceType	xs:string	An encoded string providing a canonical description of this wireless service and its associated validation rules, either implicitly via reference to an external registry or explicitly via encoded parameters in this *wirelessServiceType* attribute
name	xs:string	A human-readable name for this wireless service. For example, "Concert in the Park."

description	xs:string	A human-readable description of this wireless service. For example, “Simon and Garfunkle Concert in New York's Central Park, Sept. 1981.”
x-govtServiceId	xs:long	*Key Bridge implementation* Conveniences attribute to enable cross referencing of radio service records with corresponding Government entity records or license applications. For U.S. operation this attribute may be set to either the CDBS *application_id* or the ULS *unique_system_identifier*.
x-govtRecordDate	xs:date	*Key Bridge implementation* If this record describes a service learned from a Government-provided data source the date associated with the Government record is noted.
x-ruleNum	xs:string	*Key Bridge implementation* If this record describes a licensed service learned from a Government-provided data source and the license-associated rule is provided, that information may be recorded here.

8.19.4 Validity

At least one station is required according to the following choice of:

Station.TXRX	[1..n]	or		
Station.TX	[1..1]	plus	Station.RX	[0..n] or
Station.TX	[1..n]	plus	Station.RX	[0..1]

where

Station.TXRX is a WSIF *station* with *stationClass* attribute set to “TXRX”

Station.TX is a WSIF *station* with *stationClass* attribute set to “TX”

Station.RX is a WSIF *station* with *stationClass* attribute set to “RX”

Note that other validation rules between and among the stations within a *wirelessService* may vary according to each service’s implementation and individual characteristics.

9 Appendix: WSIF XML Schema

The WSIF is an XML-based information encoding and formatting scheme that builds and relies upon several other standard XML schema specifications. The following specifications are included by reference:

Schema	XSD Prefix	XSD Namespace URN
XSD	xs	http://www.w3.org/2001/XMLSchema
XMLDSIG	ds	http://www.w3.org/2000/09/xmldsig

9.1 Key Bridge XML Repository

The structure of WSIF elements and documents is defined using XML Schemas language. The Schema and this document are published on the Key Bridge web site at https://keybridgeglobal/2011/xml/. The main file is **wsif.xsd**.

XML schema available online at http://keybridgeglobal/2011/xml/wsif.xsd

9.2 WSIF Schema Snapshot

The following schema is provided for reference only.

THE ONLINE SCHEMA IS AUTHORITATIVE.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ds="http://www.w3.org/2000/09/xmldsig"
xmlns="http://www.keybridgeglobal.com/2011/xml/wsif.xsd"
targetNamespace="http://www.keybridgeglobal.com/2011/xml/wsif.xsd" version="1.0r31">
    <xs:complexType name="address">
        <xs:attribute name="attention"           type="xs:string"/>
        <xs:attribute name="address1"            type="xs:string"/>
        <xs:attribute name="address2"            type="xs:string"/>
        <xs:attribute name="poBox"               type="xs:string"/>
        <xs:attribute name="city"                type="xs:string"/>
        <xs:attribute name="county"              type="xs:string"/>
        <xs:attribute name="state"               type="xs:string"/>
        <xs:attribute name="postalCode"          type="xs:string"/>
        <xs:attribute name="country"             type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="antenna">
        <xs:sequence>
            <xs:element name="radiationPattern"  type="radiationPattern"/>
        </xs:sequence>
        <xs:attribute name="directional"         type="xs:boolean" use="required"/>
        <xs:attribute name="rotation"            type="xs:float" use="required"/>
        <xs:attribute name="heightAboveGround"   type="xs:float" use="required"/>
        <xs:attribute name="manufacturer"        type="xs:string"/>
        <xs:attribute name="model"               type="xs:string"/>
        <xs:attribute name="x-elevationModel"    type="xs:string"/>
        <xs:attribute name="x-govtAntennaId"     type="xs:int"/>
        <xs:attribute name="x-haat"              type="xs:float"/>
        <xs:attribute name="x-rcAmsl"            type="xs:float"/>
    </xs:complexType>
    <xs:complexType name="contact">
        <xs:sequence>
            <xs:element name="location"          type="location" minOccurs="0"/>
            <xs:element name="securityParameter" type="securityParameter" minOccurs="0"/>
            <xs:element name="vCard"             type="xs:string" minOccurs="0"/>
            <xs:element name="extension"         type="xs:string" minOccurs="0"/>
            <xs:element name="x-preferences"     type="preferences" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="uuid"                type="xs:string" use="required"/>
        <xs:attribute name="name"                type="xs:string" use="required"/>
        <xs:attribute name="email"               type="xs:string" use="required"/>
        <xs:attribute name="firstName"           type="xs:string"/>
        <xs:attribute name="lastName"               type="xs:string"/>
        <xs:attribute name="title"                  type="xs:string"/>
        <xs:attribute name="organization"           type="xs:string"/>
        <xs:attribute name="phone"                  type="xs:string"/>
        <xs:attribute name="fax"                    type="xs:string"/>
        <xs:attribute name="cell"                   type="xs:string"/>
        <xs:attribute name="contactType"            type="xs:string"/>
        <xs:attribute name="x-frn"                  type="xs:string"/>
        <xs:attribute name="x-licenseeId"           type="xs:string"/>
        <xs:attribute name="x-partyId"              type="xs:int"/>
    </xs:complexType>
    <xs:complexType name="channel">
        <xs:attribute name="allocation"             type="xs:string" use="required"/>
        <xs:attribute name="channel"                type="xs:float" use="required"/>
        <xs:attribute name="minFreq"                type="xs:double" use="required"/>
        <xs:attribute name="maxFreq"                type="xs:double" use="required"/>
    </xs:complexType>
    <xs:complexType name="enumCode">
        <xs:sequence/>
        <xs:attribute name="name"                   type="xs:string" use="required"/>
        <xs:attribute name="content"                type="xs:string" use="required"/>
        <xs:attribute name="description"            type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="location">
        <xs:sequence>
            <xs:element name="address"              type="address" minOccurs="0"/>
            <xs:element name="coordinate"           type="coordinate" minOccurs="0"/>
            <xs:element name="geometry"             type="geometry" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="uuid"                   type="xs:string" use="required"/>
        <xs:attribute name="name"                   type="xs:string" use="required"/>
        <xs:attribute name="locationType"           type="xs:string"/>
        <xs:attribute name="x-geocode"              type="xs:string"/>
        <xs:attribute name="x-haat"                 type="xs:float"/>
        <xs:attribute name="x-timeZone"             type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="coordinate">
        <xs:attribute name="latitude"               type="xs:double" use="required"/>
        <xs:attribute name="longitude"              type="xs:double" use="required"/>
        <xs:attribute name="altitude"               type="xs:double"/>
        <xs:attribute name="accuracy"               type="xs:double" use="required"/>
        <xs:attribute name="altitudeAccuracy"       type="xs:double"/>
        <xs:attribute name="heading"                type="xs:double"/>
        <xs:attribute name="speed"                  type="xs:double"/>
        <xs:attribute name="DOMTimeStamp"           type="xs:long"/>
        <xs:attribute name="x-datum"                type="xs:string"/>
        <xs:attribute name="x-elevationModel"      type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="geometry">
        <xs:sequence>
            <xs:element name="point"               type="xs:string"/>
            <xs:element name="polygon"             type="xs:string" minOccurs="0"/>
            <xs:element name="envelope"            type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="licenseInformation">
        <xs:sequence>
            <xs:element name="contact"             type="contact" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="callSign"              type="xs:string" use="required"/>
        <xs:attribute name="grantDate"             type="xs:dateTime" use="required"/>
        <xs:attribute name="expiredDate"           type="xs:dateTime" use="required"/>
        <xs:attribute name="x-dbSource"            type="xs:string"/>
        <xs:attribute name="x-eligibilityRuleNum"  type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="preferences">
        <xs:sequence>
            <xs:element name="extension"           type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="radiationPattern">
        <xs:sequence>
            <xs:element name="radiationPattern"    type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="source"                type="xs:string" use="required"/>
        <xs:attribute name="description"           type="xs:string"/>
        <xs:attribute name="interpolated"          type="xs:boolean" use="required"/>
    </xs:complexType>
    <xs:complexType name="schedule">
        <xs:sequence>
            <xs:element name="extension" minOccurs="0"/>
            <xs:element name="iCalendar" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="dtstart"               type="xs:dateTime" use="required"/>
        <xs:attribute name="dtend"                 type="xs:dateTime" use="required"/>
        <xs:attribute name="rrule"                 type="xs:string" use="required"/>
        <xs:attribute name="tzid"                  type="xs:string" use="required"/>
        <xs:attribute name="uid"                   type="xs:string" use="required"/>
        <xs:attribute name="categories"            type="xs:string"/>
        <xs:attribute name="created"               type="xs:dateTime"/>
        <xs:attribute name="classification"        type="xs:string"/>
        <xs:attribute name="description"           type="xs:string"/>
```

```
        <xs:attribute name="organizer"              type="xs:string"/>
        <xs:attribute name="priority"               type="xs:int"/>
        <xs:attribute name="summary"                type="xs:string"/>
        <xs:attribute name="transp"                 type="xs:string"/>
        <xs:attribute name="x-allDayEvent"          type="xs:boolean"/>
        <xs:attribute name="x-expirationDate"       type="xs:dateTime"/>
    </xs:complexType>
    <xs:complexType name="securityParameter">
        <xs:choice>
            <xs:element name="publicKey"            type="xs:base64Binary"/>
            <xs:element name="sharedSecret"         type="xs:base64Binary"/>
        </xs:choice>
        <xs:attribute name="protocolVersion"        type="xs:float" use="required"/>
        <xs:attribute name="securityVersion"        type="xs:float" use="required"/>
        <xs:attribute name="keyAlgorithm"           type="xs:string" use="required"/>
        <xs:attribute name="securityMode"           type="xs:string" use="required"/>
        <xs:attribute name="deviceChallenge"        type="xs:string"/>
        <xs:attribute name="deviceSecurityStatus"   type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="station">
        <xs:sequence>
            <xs:element name="channelList"          type="channel" maxOccurs="unbounded"/>
            <xs:element name="contactList"          type="contact" maxOccurs="unbounded"/>
            <xs:element name="location"             type="location"/>
            <xs:element name="antenna"              type="antenna" minOccurs="0"/>
            <xs:element name="rxStationList"        type="station" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="schedule"             type="schedule" minOccurs="0"/>
            <xs:element name="transmitterList"      type="transmitter" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="extension"            type="xs:string" minOccurs="0"/>
            <xs:element name="x-licenseInformationList"   type="licenseInformation"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="uuid"                   type="xs:string" use="required"/>
        <xs:attribute name="name"                   type="xs:string" use="required"/>
        <xs:attribute name="stationClass"           type="xs:string" use="required"/>
        <xs:attribute name="description"            type="xs:string"/>
        <xs:attribute name="stationType"            type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="transmitter">
        <xs:sequence>
            <xs:element name="channel"              type="channel" minOccurs="0"/>
            <xs:element name="extension"            type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="frequency"              type="xs:double"/>
        <xs:attribute name="deviceId"           type="xs:string"/>
        <xs:attribute name="deviceSn"           type="xs:string"/>
        <xs:attribute name="description"        type="xs:string"/>
        <xs:attribute name="ea"                 type="xs:string"/>
        <xs:attribute name="erp"                type="xs:float"/>
        <xs:attribute name="isDigital"          type="xs:boolean"/>
        <xs:attribute name="manufacturer"       type="xs:string"/>
        <xs:attribute name="model"              type="xs:string"/>
        <xs:attribute name="name"               type="xs:string"/>
        <xs:attribute name="x-digitalModRate"   type="xs:double"/>
        <xs:attribute name="x-digitalModType"   type="xs:string"/>
        <xs:attribute name="x-emissionCode"     type="xs:string"/>
        <xs:attribute name="x-equipmentClass"   type="xs:string"/>
        <xs:attribute name="x-equipmentRuleNum" type="xs:string"/>
        <xs:attribute name="x-maxErp"           type="xs:float"/>
        <xs:attribute name="x-txWidth"          type="xs:float"/>
    </xs:complexType>
    <xs:complexType name="txLPAUX">
        <xs:complexContent>
            <xs:extension base="transmitter">
                <xs:attribute name="type"          type="xs:string"/>
                <xs:attribute name="x-groupId"     type="xs:string"/>
                <xs:attribute name="x-subgroupId"  type="xs:string"/>
                <xs:attribute name="x-lectroBlock" type="xs:int"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="txTVBD">
        <xs:complexContent>
            <xs:extension base="transmitter">
                <xs:attribute name="tvbdMode"      type="xs:string" use="required"/>
                <xs:attribute name="serviceDescription"     type="xs:string"/>
                <xs:attribute name="serviceType"   type="xs:string"/>
                <xs:attribute name="x-desiredChannelList"   type="xs:string"/>
                <xs:attribute name="x-desiredFrequency"     type="xs:string"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="txTV">
        <xs:complexContent>
            <xs:extension base="transmitter">
                <xs:attribute name="callSign"      type="xs:string"/>
                <xs:attribute name="x-broadcastCallSign"    type="xs:string"/>
                <xs:attribute name="x-cntMobileUnits"       type="xs:int"/>
                <xs:attribute name="x-govtFacilityId"       type="xs:string"/>
                <xs:attribute name="x-networkAffiliation"   type="xs:string"/>
                <xs:attribute name="x-radiusOfOperation"   type="xs:int"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="wirelessService">
        <xs:sequence>
            <xs:element name="channelList"       type="channel" maxOccurs="unbounded"/>
            <xs:element name="contactList"       type="contact" maxOccurs="unbounded"/>
            <xs:element name="schedule"          type="schedule"/>
            <xs:element name="stationList"       type="station" maxOccurs="unbounded"/>
            <xs:element name="extension"         type="xs:string" minOccurs="0"/>
            <xs:element name="x-licenseInformationList"   type="licenseInformation"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="active"              type="xs:boolean" use="required"/>
        <xs:attribute name="uuid"                type="xs:string" use="required"/>
        <xs:attribute name="dateCreated"         type="xs:dateTime" use="required"/>
        <xs:attribute name="dateEffective"       type="xs:dateTime" use="required"/>
        <xs:attribute name="dateExpiration"      type="xs:dateTime" use="required"/>
        <xs:attribute name="dateLastUpdated"     type="xs:dateTime" use="required"/>
        <xs:attribute name="wirelessServiceType" type="xs:string" use="required"/>
        <xs:attribute name="name"                type="xs:string"/>
        <xs:attribute name="description"         type="xs:string"/>
        <xs:attribute name="x-govtServiceId"     type="xs:int"/>
        <xs:attribute name="x-govtRecordDate"    type="xs:dateTime"/>
        <xs:attribute name="x-ruleNum"           type="xs:string"/>
    </xs:complexType>
</xs:schema>
```

10 Appendix: Example WSIF Implementations

Key Bridge implementation

All material in this section describes Key Bridge implementations of wireless services encoded into WSIF. This material is for example only, is not authoritative, and is subject to change without notice.

10.1 A Broadcast Television WSIF *wirelessService*

The elements and attributes of a broadcast television wireless service record element are described below. Also provided below is a description of the broadcast television station element.

If the Broadcast Television wireless service is transmitting both an analog and digital signal on separate channels then one station element is present in the WSIF *wirelessService.stationList* configured as follows:

For the primary (digital) transmission:

- A WSIF *transmitter* is created and added to the *station.transmitterList* with the following configuration:
  - isDigital attribute set to true
  - *channel* element set to the digital transmission

For the secondary (analog) transmission

- A second WSIF *transmitter* is created and added to the *station.transmitterList* with the following configuration:
  - isDigital attribute set to false
  - *channel* element set to the analog transmission

Then:

- Both the analog and digital transmitter object *channel* elements are added to the *station.channelList*
- The *station.channelList* is copied to the *wirelessService.channelList*

10.1.1 Elements

Element	Documentation
channelList	Contains at least one station channel set to the value of: • the CDBS *tv_eng_data.station_channel* field if the transmitter field isDigital is set to TRUE • the CDBS *tv_eng_data.analog_channel* field if the transmitter field isDigital is set to FALST
contactList	Contains any and all contacts associated with the facility
stationList	Contains at least one station derived using information from the *tv_eng_data* and *tv_eng_data.facility* records.
schedule	Set a full-time broadcast beginning upon the call sign license grant date and ending upon the license expiration date.
x-licenseInformationList	Contains one *licenseInformation* element with the broadcast call sign and effective dates as provided in CDBS

10.1.2 Attributes

Attribute	Documentation
name	The broadcast call sign
description	Set to the a concatenation of the station's city and state as recorded in the CDBS facility.comm_city and comm._state fields (e.g. "PINE BLUFF, AR")
effectiveDate	The call sign effective date
expirationDate	The call sign expiration date
wirelessServiceType	See *Appendix: Wireless service type*

10.1.3 Validation

Broadcast Television station WSIF wirelessServices are populated according to the following content requirements:

WSIF Type	Content Rule	Parent Object	Location
channel	[1 … 2]	wirelessService	channelList
contact	[1 … n]	wirelessService	contactList
station	[1 … 2]	wirelessService	stationList
licenseInformation	[1 … 1]	wirelessService	x-licenseInformationlist

10.2 A Broadcast Television WSIF *station*

A broadcast television WSIF *station* element implementation is described below.

10.2.1 Elements

Element	Documentation
antenna	Where the antenna is directional this *antenna* element is populated with information describing the antenna make, model and radiation patter. The *antenna* element is required and may not be set to null (nullified). Where a non-directional antenna is used this *antenna* element should be populated with a normalized unity-gain antenna profile.
location	The location entity is constructed as follows: • *address* attributes are configured to the station's physical address • *coordinate* attributes are configured with coordinates in the NAD83 geodetic datum • *geometry* attributes and elements are configured as follows: o *name* attribute is typically set to the facility call sign o *locationType* attribute is set to the appropriate enumerated code o *point* element is set to a geo-location point where *x* is set to the *longitude* value and *y* is set to the *latitude* value from this location's NAD83 *coordinate* element. o *polygon* element is set to this television station's calculated geographic protected contour as defined in the Rules [27] o *envelope* element is calculated and set as a bounding rectangle of this location's *polygon* element
transmitterList	One WSIF *transmitter* object is added to the WSIF *station.transmitterList* element.
channelList	Contains ONE WSIF *channel* object set to the received channel
contactList	Contains ONE WSIF *contact* object set to the station point of contact
x-licenseInformationList	Contains ONE WSIF *licenseInformation* object corresponding to the transmitting station license information

[27] See 47 CFR 15.712(a) *Digital television stations, and digital and analog Class A TV, low power TV, TV translator and TV booster stations*

10.2.2 Attributes

Attribute	Documentation
name	Set to a concatenation the facility call sign, city and state. For example: "KVTN-DT: LITTLE ROCK, AR"
stationType	Set to "TX" to indicate a transmitting station
stationClass	Set to an enumerated code derived from the facility type For example: "us.fcc.cdbs.stationClass.DTV")
description	Set to a human-readable description of this station's *stationClass* field For example: "TV TRANSLATOR STATION VHF DIGITAL"

10.2.3 Validation

Broadcast Television station WSIF *station* objects are populated according to the following content requirements:

WSIF Type	Content Rule	Parent Object	Location
channel	[1 … 2]	station	channelList
contact	[1 … n]	station	contactList
location	[1 … 1]	station	location
antenna	[1 … 1]	station	antenna
schedule	[1 … 1]	station	schedule
station	[0 … 0]	station	stationRxList
transmitter	[1 … 2]	station	transmitterList
licenseInformation	[1 … 1]	station	licenseInformationlist

10.3 TV Translators and MVPD receive sites

10.3.1 Elements

Parameter	Description
channelList	Contains ONE element set to the received channel
contactList	Contains ONE element set to the receive *station's* contact
stationList	Contains a minimum of TWO elements: • One station is a WSIF transmitting station (stationClass=TX) corresponding to a valid, licensed broadcast television station • One or more stations are WSIF receiving stations (stationClass=TX) corresponding to the receive station
schedule	Set to a full-time broadcast receipt schedule beginning upon the WSIF wirelessService create date and extending for one (1) year.
x-licenseInformationList	For TV translators this contains ONE element corresponding to the transmitting station license information. For MVPD records this is set to null.

10.4 Urbanized Areas, including PLMRS/CMRS operations

10.4.1 Elements

Parameter	Description
channelList	Contains ONE or MORE elements set to the channel(s) enumerated in 47 CFR 90.303(a)
contactList	Empty
stationList	Contains ONE (1) WSIF receive *station* element (stationClass=RX) corresponding to the urbanized area's center-city coordinates May also contain one or more LMRS WSIF receive *station* element corresponding to any licensed services that are more than 80km from the urbanized area's center-city coordinates
schedule	Set to full-time. Urbanized areas are defined by Rule and not likely to change. For convenience and completeness, the start time is set beginning January 21, 2011 (the date white space Rules were formally adopted by the FCC) and the end time has been set to January 21, 2021 (a date sufficiently distant that no confusion is possible during any administrator's first 5-year authorization)
x-licenseInformationList	Set to null by default. If any [P,G]LMRS stations are recorded then their license information may be optionally recorded.

10.5 Offshore Radiotelephone Service

10.5.1 Elements

Parameter	Description
channelList	Contains ONE element set to the channel enumerated in 47 CFR 70.709(e)
contactList	Empty
stationList	Contains ONE station element corresponding to the ORS identified geographic area
schedule	Set to full-time. ORS areas are defined by Rule and not likely to change. For convenience and completeness, the start time is set beginning January 21, 2011 (the date white space Rules were formally adopted by the FCC) and the end time has been set to January 21, 2021 (a date sufficiently distant that no confusion is possible during any administrator's first 5-year authorization)
x-licenseInformationList	Set to null

10.6 Radio Astronomy

Parameter	Description
channelList	Contains all enumerated channels
contactList	Empty
stationList	Contains ONE WSIF receive *station* element (stationClass=RX) corresponding to the indicated geographic location and optionally geographic area
schedule	Set to full-time. See *Offshore Radiotelephone Service*
x-licenseInformationList	Set to null

10.7 Low Power auxiliary devices, including wireless microphones

Parameter	Description
channelList	Contains ONE or MORE elements set to the reserved channels of this low-power auxiliary device registration
contactList	Contains ONE or MORE elements set to a primary and possibly secondary designated contacts
stationList	Contains ONE or MORE WSIF transcieving station elements (stationClass=TXRX) corresponding to the indicated geo-location of the LPAUX station(s) or venue(s)
schedule	Contains the hours of operation and recurrence if applicable of the registered event
x-licenseInformationList	For licensed LPAUX records this element contains the license information. For unlicensed LPAUX records this element contains temporary authorization codes provided by the FCC.

11 Appendix: Enumerated Channel Values

11.1 United States

The following list represents allowable values that may be provided in a channel object for US white space operation.[28]

allocation	channel	minFreq	maxFreq
us.fcc.broadcasting.54-72	2	54	60
us.fcc.broadcasting.54-72	3	60	66
us.fcc.broadcasting.54-72	4	66	72
us.fcc.broadcasting.76-88	5	76	82
us.fcc.broadcasting.76-88	6	82	88
us.fcc.broadcasting.174-216	7	174	180
us.fcc.broadcasting.174-216	8	180	186
us.fcc.broadcasting.174-216	9	186	192
us.fcc.broadcasting.174-216	10	192	198
us.fcc.broadcasting.174-216	11	198	204
us.fcc.broadcasting.174-216	12	204	210
us.fcc.broadcasting.174-216	13	210	216
us.fcc.broadcasting.470-512	14	470	476
us.fcc.broadcasting.470-512	15	476	482
us.fcc.broadcasting.470-512	16	482	488
us.fcc.broadcasting.470-512	17	488	494
us.fcc.broadcasting.470-512	18	494	500
us.fcc.broadcasting.470-512	19	500	506
us.fcc.broadcasting.470-512	20	506	512
us.fcc.broadcasting.512-608	21	512	518
us.fcc.broadcasting.512-608	22	518	524

[28] See 47 CFR 2.106 *Table of Frequency Allocations and Radio Treaty Matters* which provides a list of allocations, TV channel number and frequency range assignments for operation within the United States

allocation	channel	minFreq	maxFreq
us.fcc.broadcasting.512-608	23	524	530
us.fcc.broadcasting.512-608	24	530	536
us.fcc.broadcasting.512-608	25	536	542
us.fcc.broadcasting.512-608	26	542	548
us.fcc.broadcasting.512-608	27	548	554
us.fcc.broadcasting.512-608	28	554	560
us.fcc.broadcasting.512-608	29	560	566
us.fcc.broadcasting.512-608	30	566	572
us.fcc.broadcasting.512-608	31	572	578
us.fcc.broadcasting.512-608	32	578	584
us.fcc.broadcasting.512-608	33	584	590
us.fcc.broadcasting.512-608	34	590	596
us.fcc.broadcasting.512-608	35	596	602
us.fcc.broadcasting.512-608	36	602	608
us.fcc.radio_astronomy.608-614	37	608	614
us.fcc.broadcasting.614-698	38	614	620
us.fcc.broadcasting.614-698	39	620	626
us.fcc.broadcasting.614-698	40	626	632
us.fcc.broadcasting.614-698	41	632	638
us.fcc.broadcasting.614-698	42	638	644
us.fcc.broadcasting.614-698	43	644	650
us.fcc.broadcasting.614-698	44	650	656
us.fcc.broadcasting.614-698	45	656	662
us.fcc.broadcasting.614-698	46	662	668
us.fcc.broadcasting.614-698	47	668	674
us.fcc.broadcasting.614-698	48	674	680
us.fcc.broadcasting.614-698	49	680	686
us.fcc.broadcasting.614-698	50	686	692
us.fcc.broadcasting.614-698	51	692	698

12 Appendix: Enumerated Codes

Note: The following enumerated codes are defined by and possibly specific to Key Bridge.

12.1 Equipment authorization agencies

Used for the *ea* attribute of the *transmitter* object.

Enumerated Code	Description	Country
us.fcc	Federal Communications Commission	USA

12.2 Contact type

Used for the *contactType* attribute of the *contact* object.

Enumerated Code Name	Description
us.ws.contactType.ADMIN	Administrative contact for a white space spectrum administrator
us.ws.contactType.ANON	A non-registered (anonymous) white space consumer
us.ws.contactType.CERTIFIER	License certifier contact on file with regulatory authority
us.ws.contactType.COORDINATOR	Designated frequency coordinator
us.ws.contactType.CUST	A registered white space customer
us.ws.contactType.EXEC	Executive contact for a white space spectrum administrator
us.ws.contactType.GENL	General contact for a white space spectrum administrator
us.ws.contactType.LICENSEE	License certifier contact on file with regulatory authority
us.ws.contactType.TECH	Technical contact for a white space spectrum administrator
us.ws.contactType.USER	A registered white space user
us.ws.contactType.WEB	A registered web portal user

12.3 Location type

In addition to the locationType attribute values derived from the FCC's CDBS and ULS databases, the following enumerated code is added as a *locationType* attribute of the location object to accommodate administrative contact records.

Enumerated Code Name	Description
us.ws.locationType.OFFICE	Office Building

12.4 Wireless service type

In addition to the *wirelessServiceType* attribute values derived from the FCC's CDBS and ULS databases, the following enumerated codes are added to accommodate white space services. The corresponding *stationType* attribute of each *wirelessServiceType* is also noted.

Enumerated Code Name	Description	Rule	Station Type
us.ws.wirelessService.ASTR	Radio Astronomy	47 CFR 15.712(h)	RX
us.ws.wirelessService.BAS	Broadcast Auxiliary Service	47 CFR 15.712(c)	TXRX
us.ws.wirelessService.LMRS	Land Mobile Wireless Service	47 CFR 15.712(d)	TXRX
us.ws.wirelessService.LPAUX	Low Power Auxiliary Station	47 CFR 15.712(f)	TXRX
us.ws.wirelessService.MVPD	Multichannel Video Programming Distributor	47 CFR 15.712(b)	TXRX
us.ws.wirelessService.ORS	Offshore Radiotelephone Service	47 CFR 74.709(e)	TXRX
us.ws.wirelessService.TV	Broadcast Television	47 CFR 15.712(a)	TX
us.ws.wirelessService.TVBD	TV Band Device (Unlicensed)	47 CFR 15.713(2)(vi)	TX
us.ws.wirelessService.TVTX	Television Translator and LP/MVPD Receive Station	47 CFR 15.712(b)	TXRX
us.ws.wirelessService.URBAN	LMRS Reserved Urbanized Area	47 CFR 90.303	TXRX

12.5 Station type

Allowed values of the *stationType* attribute for the WSIF *station* object are enumerated below:

Enumerated Code Name	Description	Station Type
us.ws.stationType.TX	Transmit Station	TX
us.ws.stationType.RX	Receive Station	RX
us.ws.stationType.TXRX	Transmit and Receive Station	TXRX

13 Appendix: SFA Geometry Class Hierarchy and Encoding[29]

The geometric coding employed by the Wireless Service Information Format relies upon a subset of classes defined in the Simple Feature Access (SFA) Geometry model. These are *Point*, *Polygon* and *Multipoint*, which are themselves implementations of the SFA *Geometry* object interface and, by definition, may include a SFA *SpatialReferenceSystem* and a SFA *MeasureReferenceSystem*.

In the WSIF specification only a *SpatialReferenceSystem* is allowed, which is used to identify and persist the geodetic datum.

Any *MeasureReferenceSystem* should be ignored.

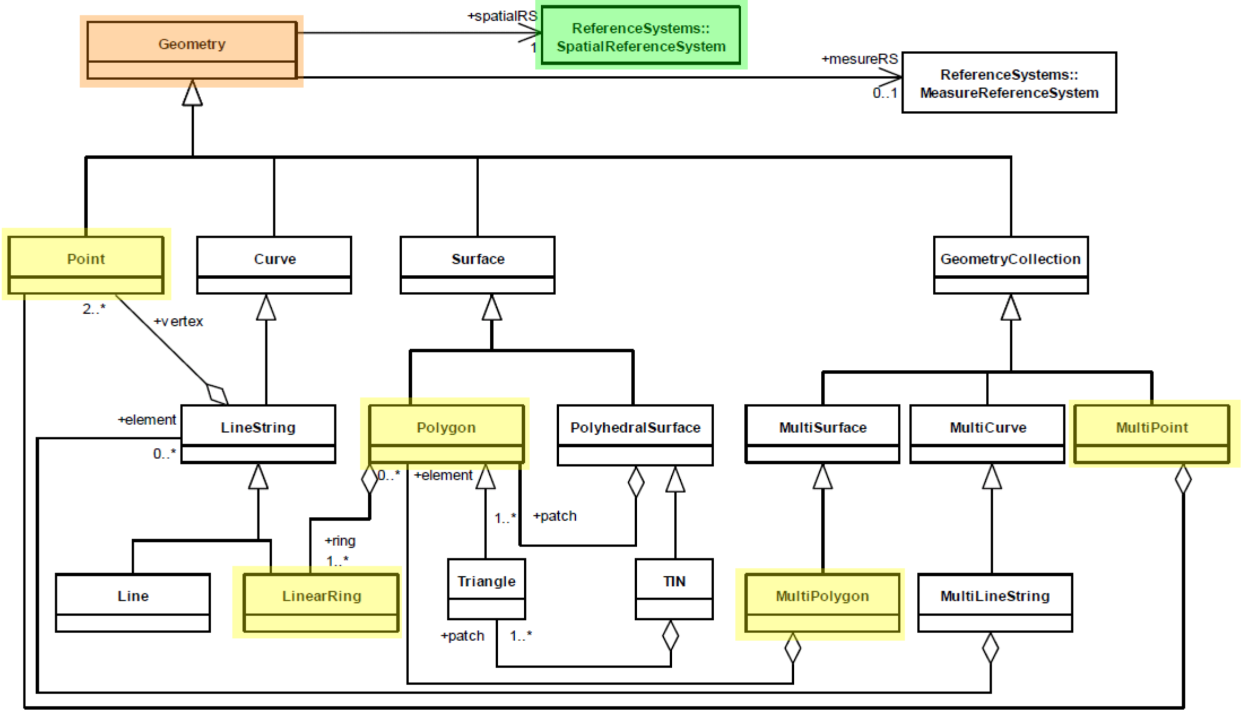


Figure 23: Geometry class hierarchy showing the relationship between Point, LinearRing, Polygon. MultiPoint and MultiPolygon objects (yellow). All objects within this specification are encoded (and exchanged) as Geometry objects (orange) and must be inspected to determine their class type. Geometry objects should always identify their spatial reference system (green).

[29] Definitions in this section are lightly edited from the OpenGIS® *Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture* and are Copyright © 2010 Open Geospatial Consortium, Inc.

13.1.1.1 SFA Point

A Point is a 0-dimensional geometric object and represents a single location in coordinate space. A Point has an x-coordinate value (longitude) and a y-coordinate value (latitude). If allowed for by the associated Spatial Reference System, it may also have coordinate values for z (altitude).

13.1.1.2 SFA MultiPoint

A MultiPoint is a 0-dimensional GeometryCollection. The elements of a MultiPoint must be Points which are not connected nor ordered in any semantically important way.

13.1.1.3 SFA LineString, Line, LinearRing

A LineString is a Curve with linear interpolation between Points as shown in *Figure 24: item (a)*. Each consecutive pair of Points defines a Line segment (b). A Line is a LineString with exactly 2 Points.

A LinearRing is a LineString that is both closed and simple. The Curve in item (c) is a closed LineString that is a LinearRing. The Curve shown in item (d) is a closed LineString that is not a LinearRing.

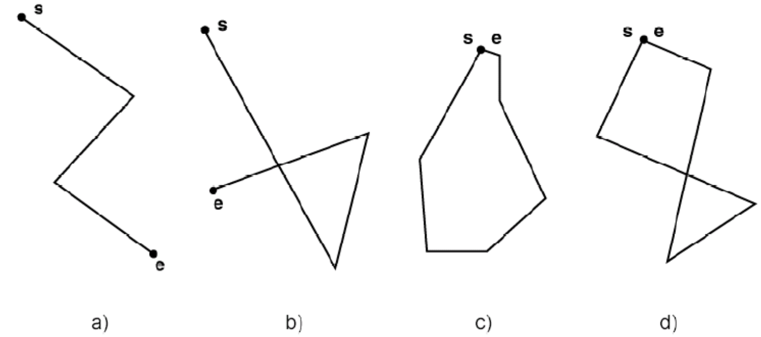


Figure 24: LineString examples. s = start, e = end.

13.1.1.4 SFA Surface

A "Surface" is defined as a 2-dimensional geometric object.

A simple Surface may consists of a single "patch" that is associated with one "exterior boundary" and 0 or more "interior" boundaries, and polyhedral Surfaces are formed by "stitching" together such simple Surfaces patches along their common boundaries.

The only instantiable subclass of Surface used by the WSIF standard is *Polygon*. A Polygon is a simple Surface that is planar.

13.1.1.5 SFA Polygon

A Polygon is a planar Surface defined by 1 exterior boundary (defined by a closed LinearRing) and 0 or more interior boundaries (also defined by one or more closed LinearRings). Each interior boundary defines a hole in the Polygon. The exterior boundary LinearRing defines the "top" of the surface as is the side of the surface from which the exterior boundary appears to traverse the boundary in a counter clockwise direction (e.g. following the right-hand-rule [30]). Interior LinearRings will have the opposite orientation, and appear as clockwise when viewed from the "top",

The rules that define valid Polygons are as follows:

a) Polygons are topologically closed (i.e. the exterior boundary is by a closed LinearRing);

b) The boundary of a Polygon consists of a set of LinearRings that make up its exterior and interior boundaries;

c) No two Rings in the boundary cross; boundary Rings may intersect at a Point but only as a tangent

d) A Polygon may not have cut lines, spikes or punctures

e) The interior of every Polygon is a connected point set;

f) The exterior of a Polygon with 1 or more holes is not connected; that is, each hole defines a connected component of the exterior.

The combination of (a) and (c) makes a Polygon a regular closed Point set and a simple geometric objects.

Figure 25 illustrates several examples of Polygons while Figure 26 illustrates examples of geometric objects that violate the Polygon assertions and are not valid instances.

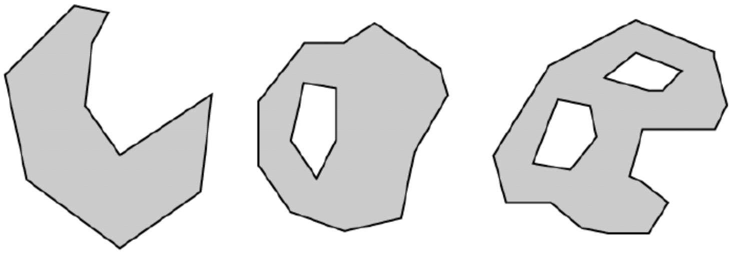


Figure 25: Valid Polygons

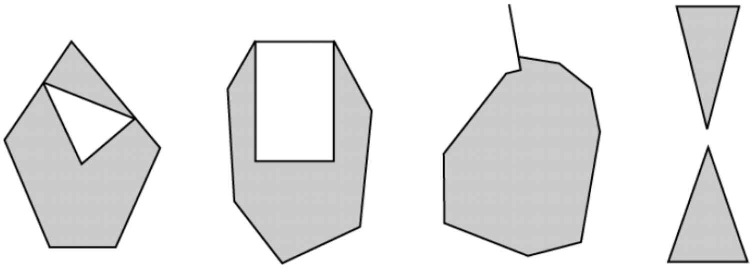


Figure 26: Invalid Polygon instances due to a violation of one or more assertion.

[30] In mathematics and physics the right-hand rule is a common mnemonic used to indicate the direction of vector multiplication in 3 dimensions. Online at http://en.wikipedia.org/wiki/Right-hand_rule.

13.1.1.6 SFA MultiPolygon

A MultiPolygon is a MultiSurface whose elements are Polygons. The assertions for MultiPolygons are:

a) The interiors of 2 Polygons that are elements of a MultiPolygon may not intersect.

b) The boundaries of any 2 Polygons that are elements of a MultiPolygon may not “cross” and may touch at only a finite number of Points.

c) A MultiPolygon is defined as topologically closed.

d) A MultiPolygon may not have cut lines, spikes or punctures; a MultiPolygon is a regular closed Point set:

e) The interior of a MultiPolygon with more than 1 Polygon is not connected; the number of connected components of the interior of a MultiPolygon is equal to the number of Polygons in the MultiPolygon.

The boundary of a MultiPolygon is a set of closed Curves (LineStrings) corresponding to the boundaries of its element Polygons. Each Curve in the boundary of the MultiPolygon is in the boundary of exactly 1 element Polygon, and every Curve in the boundary of an element Polygon is in the boundary of the MultiPolygon.

Figure 17 shows four examples of valid MultiPolygons with 1, 3, 2 and 2 Polygon elements, respectively.

Figure 18 shows examples of geometric objects not valid MultiPolygon instances.

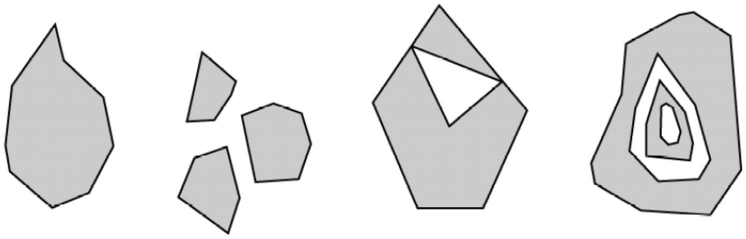


Figure 27: Valid MultiPolygon instances.

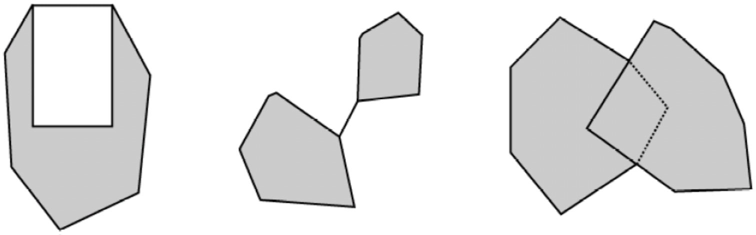


Figure 28: Invalid MultiPolygon instances due to a violation of one or more assertion.

14 Appendix: Encoding of WSIF Geographic Information

Key Bridge implementation

Geo-location and geographic information are encoded, published and distributed according to the Simple Feature Access (SFA) Well-Known-Text (WKT) specification described in *Appendix: SFA Geometry Class Hierarchy and* Encoding.[31]

All geographic and geo-location information within this specification, and specifically as relates to the WSIF *geometry* object, is communicated according to the following geographic information encoding strategy:

Service Type	Geometry Encoding
Broadcast Television	Protected service contours are communicated as a SFA WKT Polygon implementation, wherein the co-channel geographic protected contour is encoded into a closed LineString and set as the Polygon's boundary.
Broadcast Auxiliary Service	Protected service contours are communicated as a SFA WKT Polygon implementation, wherein co-channel protected service contours are encoded into a closed LineString and set to the Polygon's boundary while the adjacent-channel protected contour is encoded into a separate closed LineString and set within the Polygon's interior (e.g. as an interior boundary)
TV Translators and MVPD receive sites	TV Translator and MVPD contours are encoded according to the same procedure as Broadcast Auxiliary Service contours.
Urbanized Areas, including PLMRS/CMRS operations	Protected service contours are communicated as a SFA WKT Polygon implementation, wherein a circular closed LineString is calculated around each enumerated Urbanized Area center and the two protected contours (co- and adjacent channel) are is encoded into a single Polygon as follows: • The co-channel protected contour is encoded as a closed LineString and set to the Polygon's boundary • The adjacent channel protected contour is encoded into a closed LineString and set within the co-channel Polygon's interior. • If a LPRS contour extends outside the main area, that LMRS contour is joined, via a geometric union, with the Urbanized area contour to produce a new, extended protected contour. Note that individual contours may still be reviewed by investigating stations in the WSIF *wirelessService.stationList* element.
Offshore Radiotelephone Service	Protected service contours are communicated as a SFA WKT Polygon implementation, wherein the boundaries of each defined area are encoded as a closed LineString and set to the

[31] For non-public transactions Key Bridge also supports the exchange of geolocation and geographic information according the SFA well-known-binary (WKB) specification.

Polygon’s boundary.

Service Type	Geometry Encoding
Radio Astronomy	Protected service contours are communicated as a WFA WKT Polygon implementation, wherein a circular closed LineString is calculated around each Radio Astronomy site and set to the Polygon’s boundary.
Low Power auxiliary devices, including wireless microphones	Protected service contours are communicated as a Geometry object and must be inspected by the recipient to determine the implementation.

15 Appendix: XPath Syntax

XPath is a language used to search and query an XML document. XPath is an important element in XSLT and is a W3C Standard. XPath uses path expressions to select element nodes or node-sets in an XML document. XPath includes over 100 built-in functions, with accommodation for string values, numeric values, date and time comparison, node manipulation, sequence manipulation, Boolean values, and more.

For purposes of discussion, the following simple XML document is referred to:

```
<bookstore>
    <book>
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
         <price>29.99</price>
    </book>
</bookstore>
```

Example of nodes in the example XML document are:

<bookstore>	(document node)
<author>J K. Rowling</author>	(element node)
lang="en"	(attribute node)

15.1 XPath Nodes

In XPath, there are seven kinds of nodes: element, attribute, text, namespace, processing-instruction, comment, and document (root) nodes. XML documents are treated as trees of nodes. The root of the tree is called the document node (or root node).

XPath defines the following terms:

Term	Definition
Atomic values	Atomic values are nodes with no children or parent.
Items	Items are atomic values or nodes.
Parent	Each element and attribute has one parent. In the example the book element is the parent of title, author, year, and price.
Children	Element nodes may have zero, one or more children. In the example the title, author, year, and price elements are all children of book.
Siblings	Nodes that have the same parent. In the example title, author, year, and price are all siblings.
Ancestors	A node's parent, parent's parent, etc. In the example the ancestors of title are book and bookstore.
Descendants	A node's children, children's children, etc. In the example descendants of bookstore are book, title, author, year, and price.

15.2 XPath expressions

XPath uses path expressions to select nodes or node-sets in an XML document. The node is selected by following a path or steps. The most useful path expressions are listed below:

Expression	Description
nodename	Selects all child nodes of the named node
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@attributename	Selects attributes

15.3 XPath predicates

Predicates are used to find a specific node or a node that contains a specific value. They appear after the node they qualify and are embedded in square brackets.

Examples of predicates (based on the example XML document above):

Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element.[32]
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element
//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='eng']	Selects all the title elements that have an attribute named lang with a value of 'eng'
/bookstore/book[price>35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

[32] The W3C standard identifies [1] as the first XPath node, however some implementations may differ. For example, XPath implementations in Microsoft Internet Explorer 5 and later identify [0] as the first node. WSIF uses the W3C standard of [1] as the first XPath node.

15.4 Selecting unknown nodes

XPath wildcards can be used to select unknown XML elements.

Wildcard Expression	Description
*	Matches any element node
/bookstore/*	Selects all the child nodes of the bookstore element
@*	Matches any attribute node
//title[@*]	Selects all title elements which have any attribute
node()	Matches any node of any kind

15.5 Compound search

By using the pipe ("|") operator in an XPath expression you can combine multiple search criterion into a single search query.

Path Expression	Result
//book/title | //book/price	Selects all the title AND price elements of all book elements
//title | //price	Selects all the title AND price elements in the document
/bookstore/book/title | //price	Selects all the title elements of the book element of the bookstore element AND all the price elements in the document

A location path can be absolute or relative. An absolute location path starts with a slash (/) and a relative location path does not. In both cases the location path consists of one or more steps, each separated by a slash. For example, an absolute location path would look like: /step/step/... while a relative location path does not have the beginning slash: step/step/....